



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

AULA 05 – CHATGPT I

# O QUE IREMOS APRENDER

**01**

FERRAMENTAS DE IA

**02**

INTRODUÇÃO AO CHATGPT

**03**

CHATGPT PARA PROGRAMAÇÃO

**04**

REVISÃO ESTRUTURAS CONDICIONAIS (EXERCÍCIOS)

**05**

REVISÃO ESTRUTURAS DE REPETIÇÃO (EXERCÍCIOS)

# Ferramentas de IA

A partir de 2023 o uso das inteligências artificiais começaram a ser cada vez mais utilizadas. Entre elas, o **ChatGPT** com certeza foi a que mais se destacou.

Atualmente existem diversas ferramentas de inteligência artificial que facilitam muito a nossa vida, seja **automatizando processos**, analisando dados ou **gerando textos e imagens** em questão de segundos.



# Ferramentas de IA

Linguagem  
Natural



ChatGPT

Geração de  
Imagem



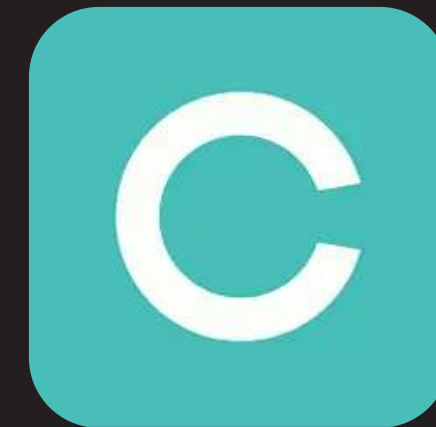
Midjourney

Geração de  
Imagem



Leonardo AI

Marketing e  
Textos



Copy AI

# Introdução ao ChatGPT

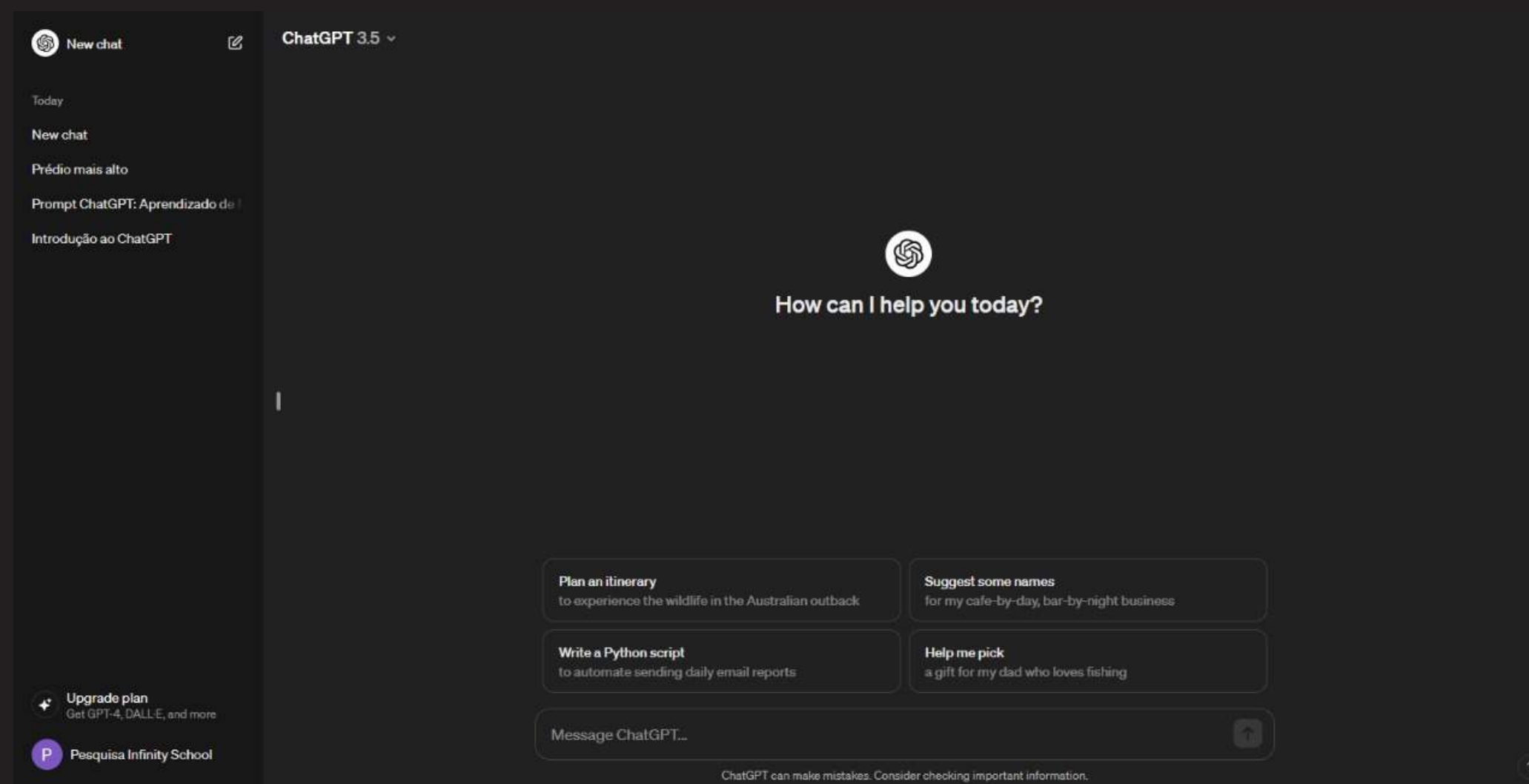


O **ChatGPT** é uma inteligência artificial desenvolvida pela **Open AI**, sendo uma inteligência artificial feita para **processamento de linguagem natural** que foi treinada com o conteúdo da internet. O seu funcionamento se baseia na criação de **chats**, onde nós informamos um determinado **prompt** 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. Onde podemos criar os **chats**, para conversar com o chatgpt.

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





# ATIVIDADE PRÁTICA

## **Atividade 01**

Crie uma conta no ChatGPT no site da open.ai.

## **Atividade 02**

Crie um novo chat, e faça 3 perguntas acerca de um assunto que você tem curiosidade na programação.

**(prompt)** > “Me explique com exemplos como a lógica de programação será utilizada para construir sites e porque ela é tão importante.

# Condicionais

As condicionais nos permitem **tomar decisões** dentro do código, considerando **condições** para que algo aconteça a partir da seguinte lógica:

**SE <condição>**  
**ENTÃO faça x**  
**SENÃO faça y**

Lembrando que nossas condições serão definidas pela **comparação** de dois valores, utilizando os **operadores relacionais**. Também podemos utilizar os **operadores lógicos** para comparar duas ou mais condições.





# Condicionais

Utilização de uma estrutura condicional



```
1  n1 = int(input('Digite um numero: '))
2  n2 = int(input('Digite outro numero: '))
3  if n1 > n2:
4      print(f'O numero {n1} é maior')
5  else:
6      print(f'O numero {n2} é maior')
```

# Condicionais

Perceba que, na estrutura anterior, ainda temos um problema: e se os *números forem iguais*? A estrutura condicional cairá no **else**, pois a nossa condição será falsa.

Nesses casos que temos mais de duas situações possíveis, precisamos adicionar o “**SENÃO SE**” que vai possibilitar abranger mais decisões possíveis a serem tomadas.

**SE <condição>**  
ENTÃO faça x  
**SENÃO SE <condição>**  
ENTÃO faça y  
**SENÃO** faça z



```
n1 = int(input('Digite um numero: '))
n2 = int(input('Digite outro numero: '))
if n1 > n2:
    print(f'O numero {n1} é maior')
elif n2 > n1:
    print(f'O numero {n2} é maior')
else:
    print('Os numeros são iguais')
```

# ATIVIDADE PRÁTICA

## Atividade 03

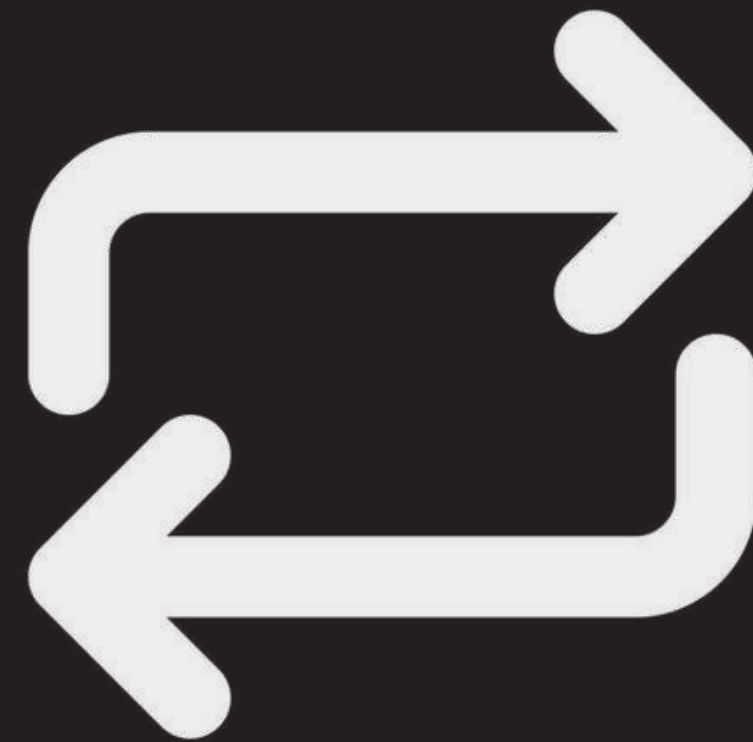
Peça para que o ChatGPT gere para você 5 exercícios de condicionais

**(prompt)** > “Faça para mim 5 exercícios de Lógica de programação para praticar estruturas condicionais . Os exercicios devem ser personalizados, considerando que eu... {algun contexto sobre sua vida}”

# Estruturas de Repetição

Uma **estrutura de repetição**, também conhecida como **loop**, é uma estrutura que permite que a gente **repita uma ou mais instruções várias vezes**.

A quantidade de repetições pode ser definida por uma **quantidade** ou por uma **condição**. O que definirá se você utilizará o **FOR** ou **WHILE**.



# Loop: For

A estrutura de repetição **FOR** nos permite repetir um determinado bloco de código uma **quantidade determinada de vezes**. Exemplo:

PARA CADA VALOR DE **«variável»** NO INTERVALO DE **1 à 10**  
FAÇA: MOSTRE **«variável»**



```
1  for variavel in range(1, 11):  
2      print(variavel)
```

Final Exclusivo (n - 1)



# Loop: For

## Como ocorre a repetição?

1ª ITERAÇÃO | VARIÁVEL = 1 & MOSTRA VARIÁVEL

2ª ITERAÇÃO | VARIÁVEL = 2 & MOSTRA VARIÁVEL



9ª ITERAÇÃO | VARIÁVEL = 9 & MOSTRA VARIÁVEL

10ª ITERAÇÃO | VARIÁVEL = 10 & MOSTRA VARIÁVEL

# 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
```

# ATIVIDADE PRÁTICA

## **Atividade 04**

Faça um programa que receba uma palavra e mostre somente as vogais.

## **Atividade 05**

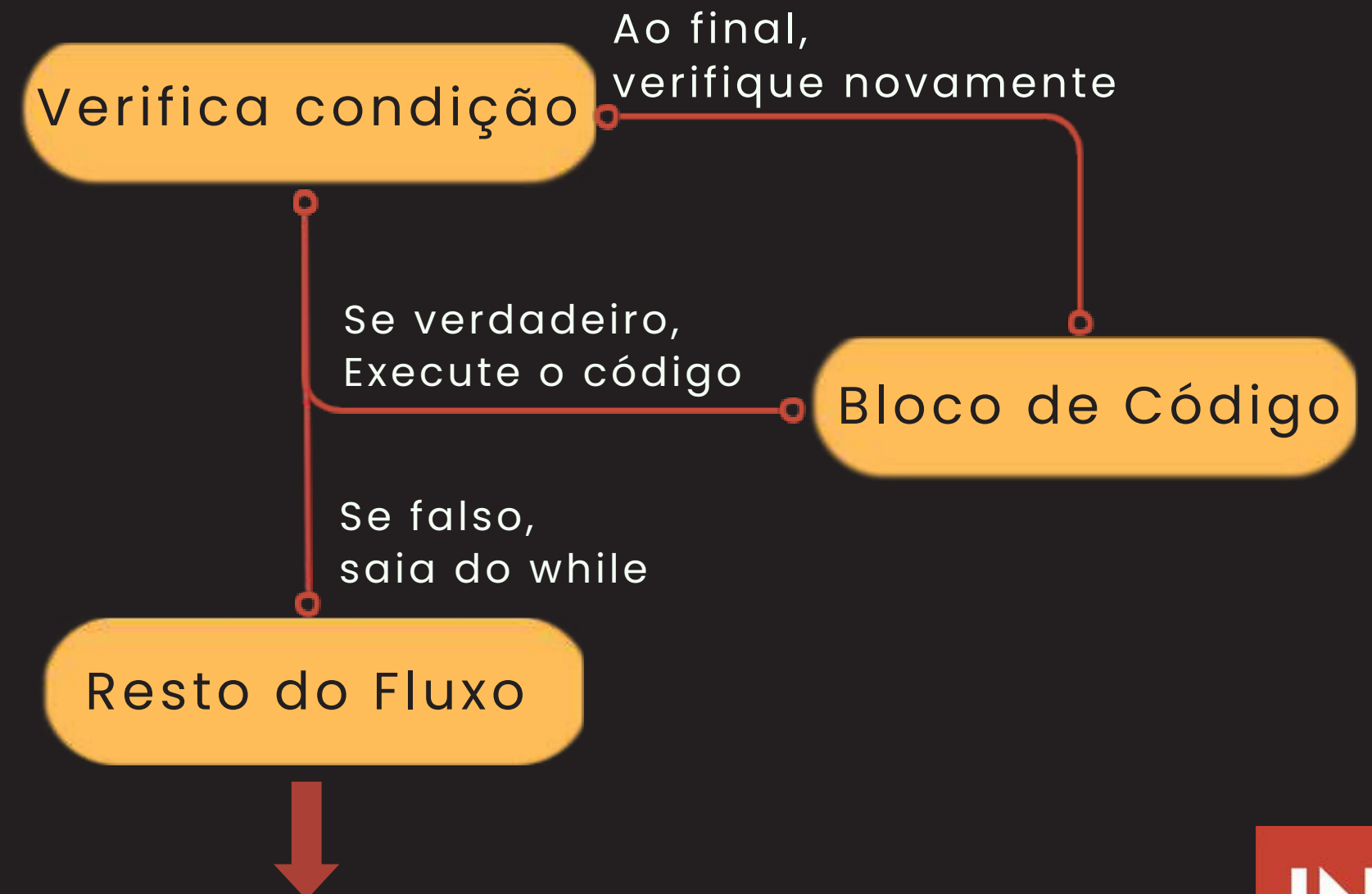
Faça um programa que peça 5 números e mostre o maior entre eles.

# 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
```



# Loop: While

Quando estamos utilizando o **WHILE**, precisamos de uma **variável de controle** para gerenciar **quando a repetição irá parar**. É necessário, também, que **alteremos a nossa variável de controle** em algum momento dentro do bloco de código do **WHILE** para que a condição se torne falsa, ou podemos ficar presos em um loop infinito.



```
1  contador = 1  ← Variável Controle
2  while contador < 11:
3      print(contador)
4      contador += 1  ← Alterando o controle
```



# 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)
```

# ATIVIDADE PRÁTICA

## **Atividade 06**

Faça um programa que receba 3 números e, ao final, mostre a soma deles (utilizando o **WHILE**).

## **Atividade 07**

Faça um programa em que o usuário só possa digitar valores numéricos. Caso ele digite uma letra, mostre uma mensagem de erro e peça novamente um valor.



# SE LIGA NO CONTEÚDO DA PRÓXIMA AULA!

AULA 06 DE LÓGICA DE PROGRAMAÇÃO:  
CHATGPT II

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

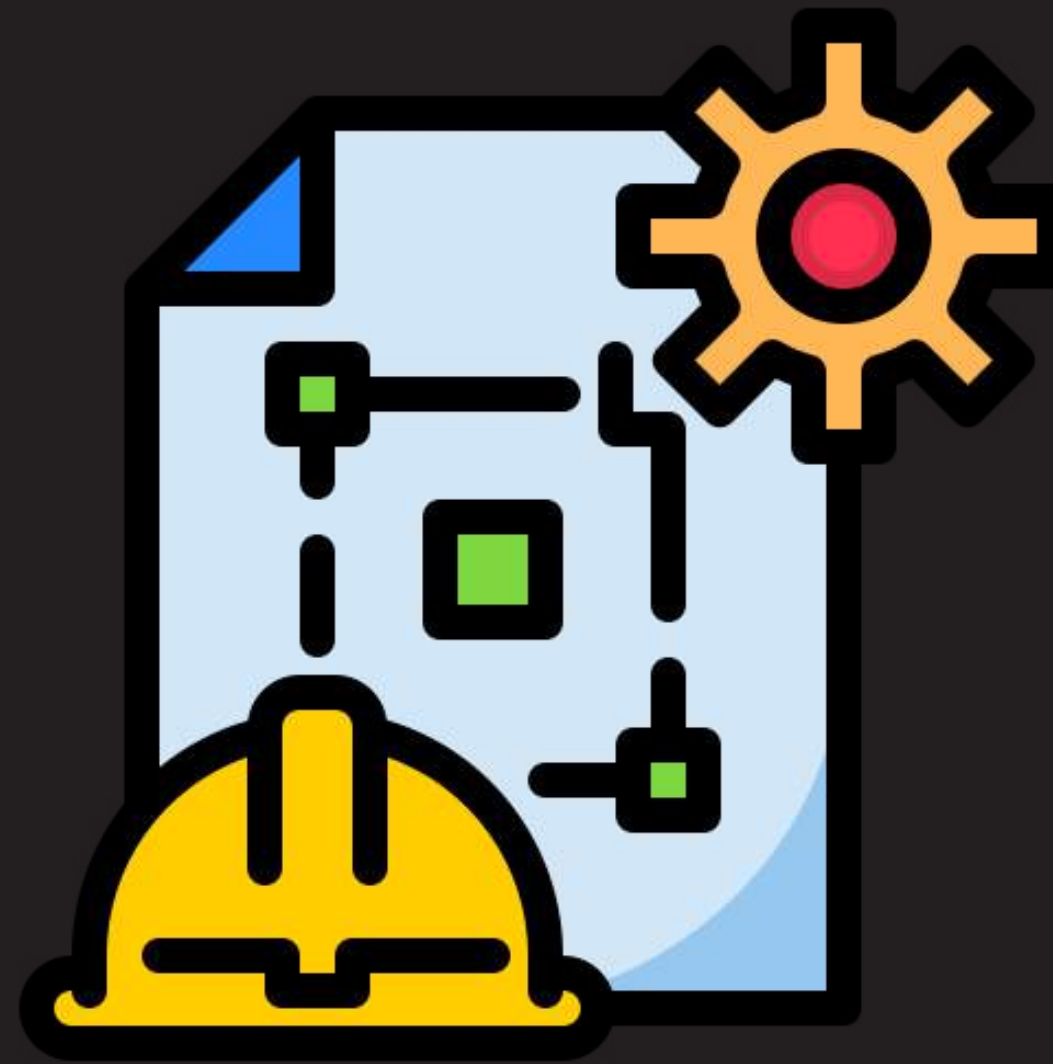
**INFINITY SCHOOL**  
VISUAL ART CREATIVE CENTER



# Vamos aprender sobre...

**ENGENHARIA DE PROMPT**

**OUTRAS IAS**



PHIND



PERPLEXITY



CHATGPT

IN



INFINITY SCHOOL

VISUAL ART CREATIVE CENTER

ATÉ A PRÓXIMA!