

Sistema de Computação



Sistema de computação é um conjunto de componentes e dispositivos interconectados que trabalham juntos para realizar tarefas específicas. Ele é composto por hardware, software e dispositivo de entrada e saída. O hardware inclui componentes como cpu, memória, armazenamento, dispositivo de entrada e saída, enquanto o software inclui sistemas operacionais e aplicativos.

Principais Componentes

Um dos principais componentes de um sistema de computação é o processador, também conhecido como CPU. Ele é responsável por executar as instruções do software e controlar as operações do sistema. A memória é outro componente importante, ela armazena dados temporariamente enquanto o processador os está processando.

Armazenamento

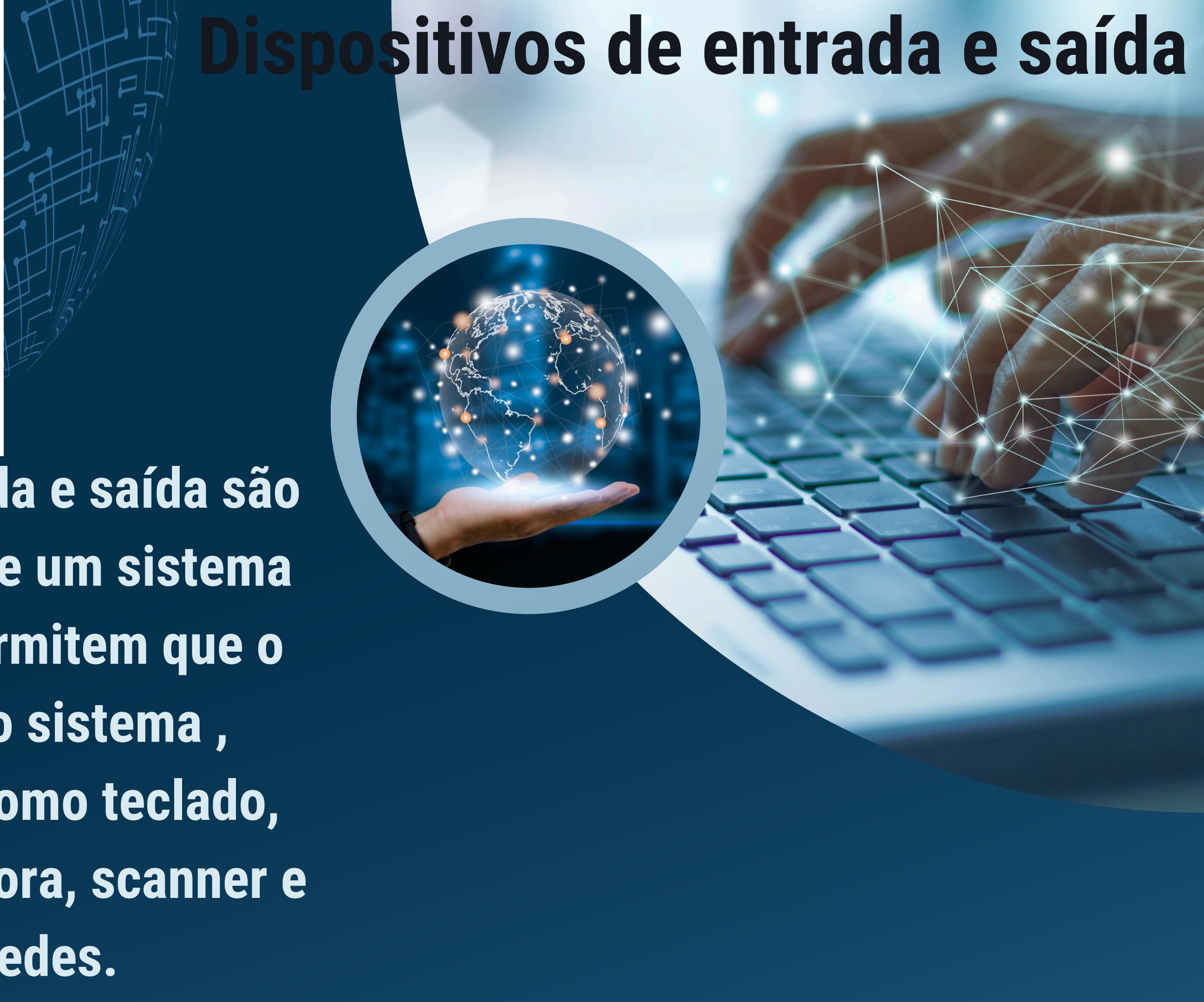
O armazenamento é outro componente importante de um sistema de computação, ele é responsável por armazenar os dados permanentemente, incluindo sistemas operacionais, aplicativos e arquivos. Existem vários tipos de dispositivo de armazenamento, como discos rígidos, unidades flash e disco externo.



Dispositivos de entrada e saída



Os dispositivos de entrada e saída são outra parte importante de um sistema de computação. Eles permitem que o usuário interaja com o sistema , incluindo dispositivos como teclado, mouse, monitor, impressora, scanner e dispositivos de redes.



ORGANIZAÇÃO DE COMPUTADORES



CPU: O processador é um dispositivo eletrônico fundamental para o funcionamento de computadores. Ele é responsável por realizar cálculos, armazenar informações e executar instruções de programas. É a principal unidade de processamento de dados de um computador.

CPU

CORE

vCPU

Memórias: memória de computador é o local onde o computador armazena os dados temporários ou permanentes que são necessários para o seu funcionamento. Existem vários tipos de memórias, incluindo memória RAM, memória ROM, memória FLASH e memória de vídeo.

Memórias RAM: é uma forma de armazenamento temporário de dados em um computador. Ela é usada para armazenar dados e programas que estão sendo usadosativamente pelo processador. A memória RAM é volátil, o que significa que os dados são perdidos quando o computador é desligado. A principal função da RAM é permitir que o processador accesse rapidamente os dados necessários para executar tarefas.



Memórias ROM: é um tipo de armazenamento de dados em um dispositivo eletrônico que retém suas informações mesmo quando a energia é desligada. Ao contrário da RAM, a ROM não é volátil, o que significa que os dados nela armazenados são permanentes e não podem ser facilmente alterados ou apagados.

Memórias Flash: é um tipo de memória não volátil que é eletricamente apagável e regravável. Ela é comumente usada em dispositivos eletrônicos para armazenamento de dados, como em unidades USB, cartões de memórias, SSDs e alguns dispositivos móveis como smartphones e tablets.

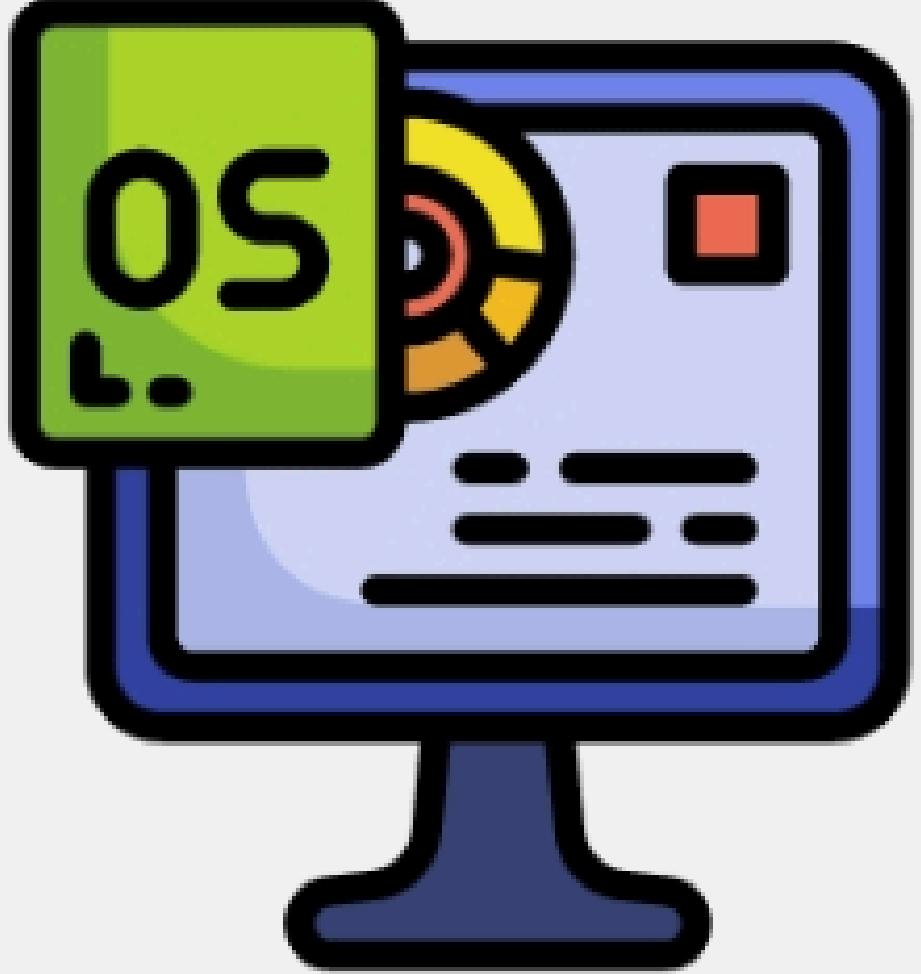


Circuitos Lógicos: um circuito lógico é um conjunto de dispositivos eletrônicos interligados por meio de conexões elétricas, que executam operações lógicas. O objetivo de um circuito lógico é criar uma saída lógica, baseada nos dados de entrada.

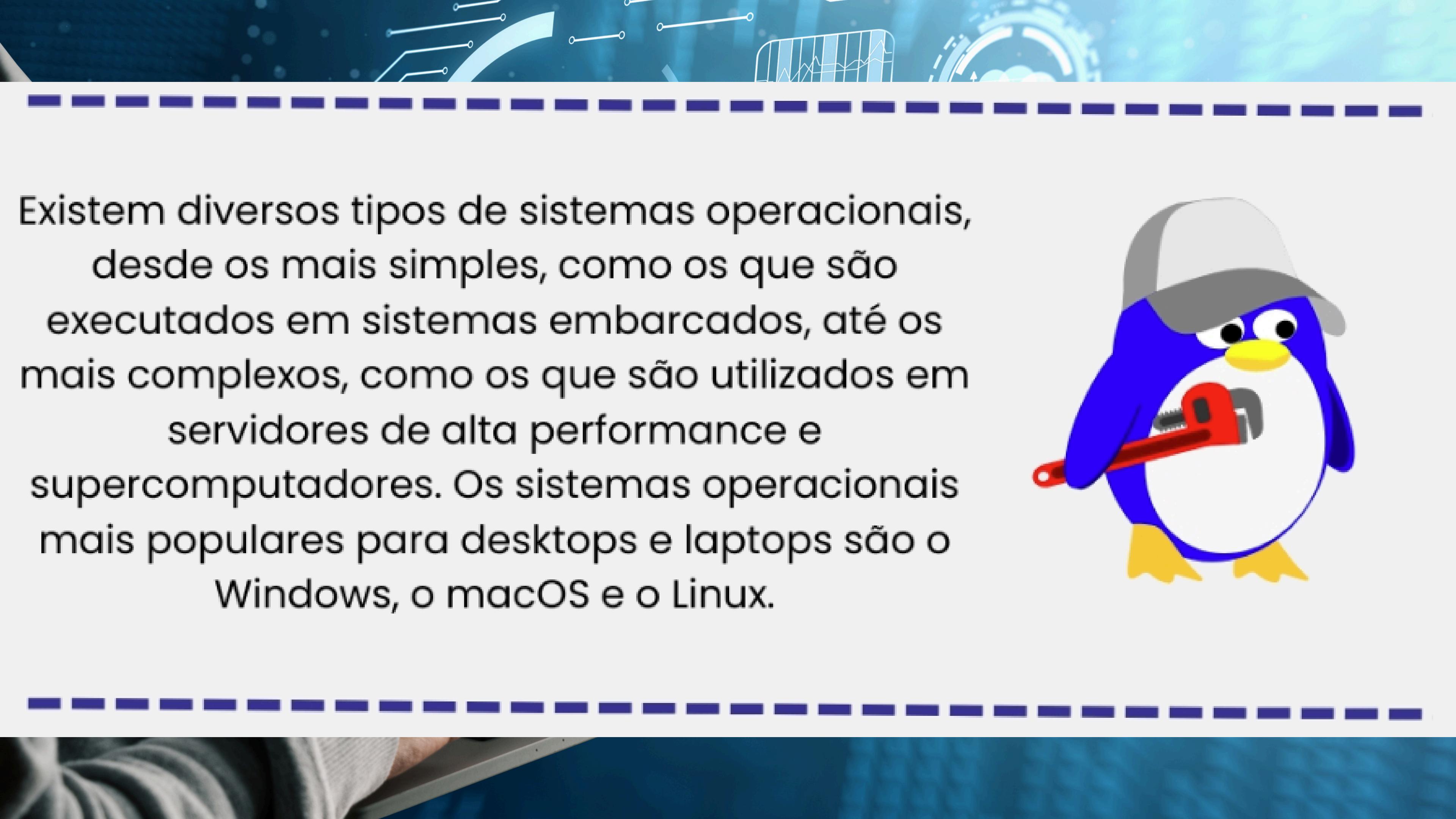
Software Básico: O software é um conjunto de instruções que permite que um computador ou dispositivo eletrônico execute tarefas específicas. É programado em linguagem de computador e pode ser usado para executar várias tarefas, desde a criação de documentos e planilhas até a execução de jogos. O software também pode ser usado para controlar outros dispositivos, uma máquina de fabricação automatizada ou equipamentos de telecomunicação.

macOS Big Sur

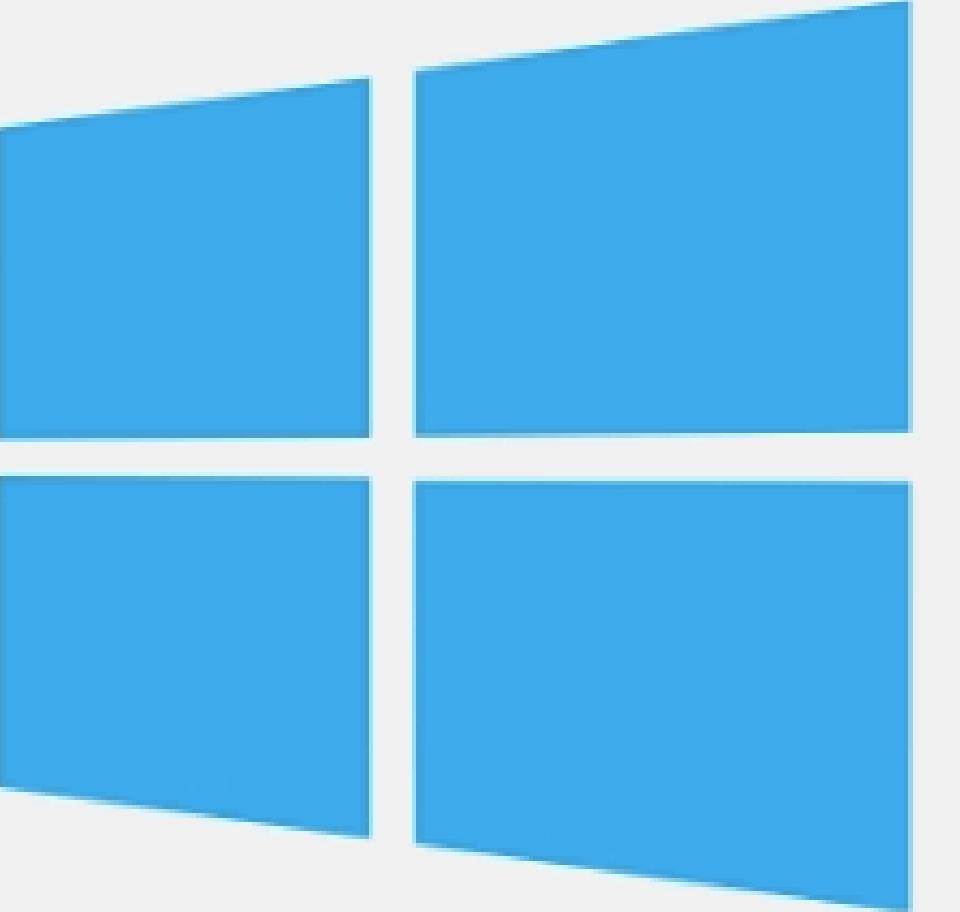
Sistemas operacionais



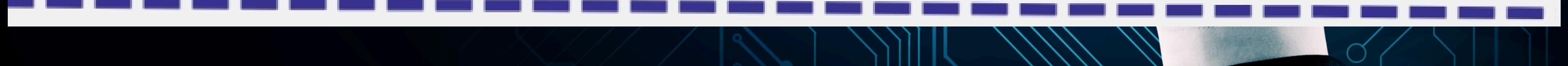
Um sistema operacional é um software que gerencia os recursos de hardware e software de um computador, permitindo que os aplicativos possam ser executados e tenham acesso a esses recursos. Ele é responsável por administrar a utilização do processador, da memória, dos dispositivos de entrada e saída, dos arquivos do sistema, além de fornecer uma interface com o usuário.



Existem diversos tipos de sistemas operacionais, desde os mais simples, como os que são executados em sistemas embarcados, até os mais complexos, como os que são utilizados em servidores de alta performance e supercomputadores. Os sistemas operacionais mais populares para desktops e laptops são o Windows, o macOS e o Linux.



O sistema operacional atua como uma camada intermediária entre o hardware e o software, permitindo que os aplicativos sejam executados de forma eficiente e segura. Ele gerencia o acesso aos recursos do hardware, garantindo que múltiplos aplicativos possam compartilhar esses recursos sem interferir uns com os outros. Além disso, ele fornece uma camada de segurança, protegendo o sistema contra malware e outras ameaças.



O sistema operacional é responsável por administrar a instalação e remoção de programas, bem como por garantir que o sistema esteja sempre atualizado e seguro. Ele também fornece uma interface com o usuário, permitindo que as pessoas possam interagir com o computador de forma amigável e intuitiva.

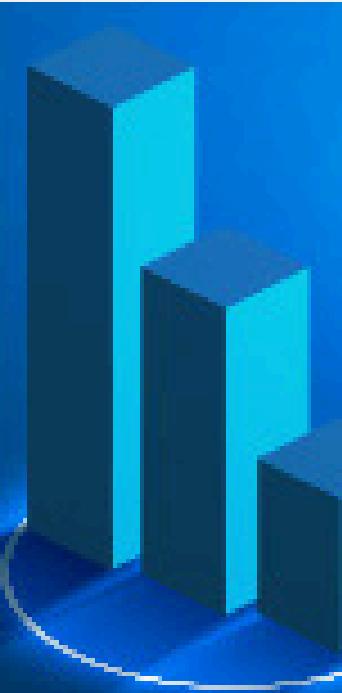


o sistema operacional é um componente essencial de qualquer computador, permitindo que os aplicativos possam ser executados e os recursos do hardware possam ser utilizados de forma eficiente. Ele é responsável por gerenciar a utilização do processador, da memória, dos dispositivos de entrada e saída, dos arquivos do sistema e fornecer uma interface com o usuário, além de fornecer uma camada de segurança para proteger o sistema contra ameaças externas.

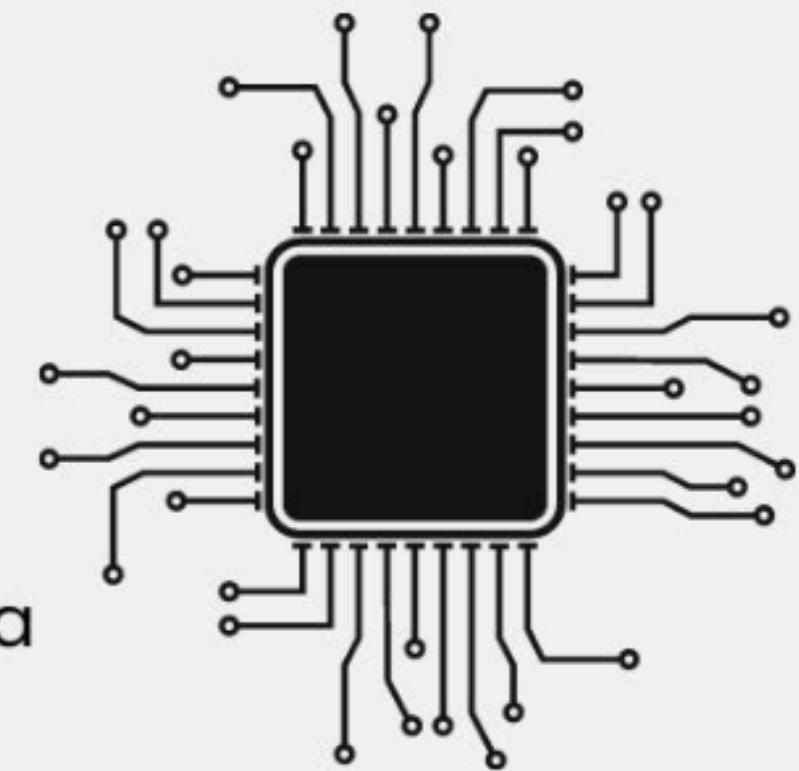
Estrutura interna do sistema operacional



A estrutura interna de um sistema operacional é composta por várias camadas que trabalham juntas para fornecer uma interface de usuário amigável e um ambiente seguro e estável para executar aplicativos. Essas camadas incluem o núcleo, o gerenciador de processos, o gerenciador de memória, o sistema de arquivos, o gerenciador de entrada/saída e os drivers de dispositivo.



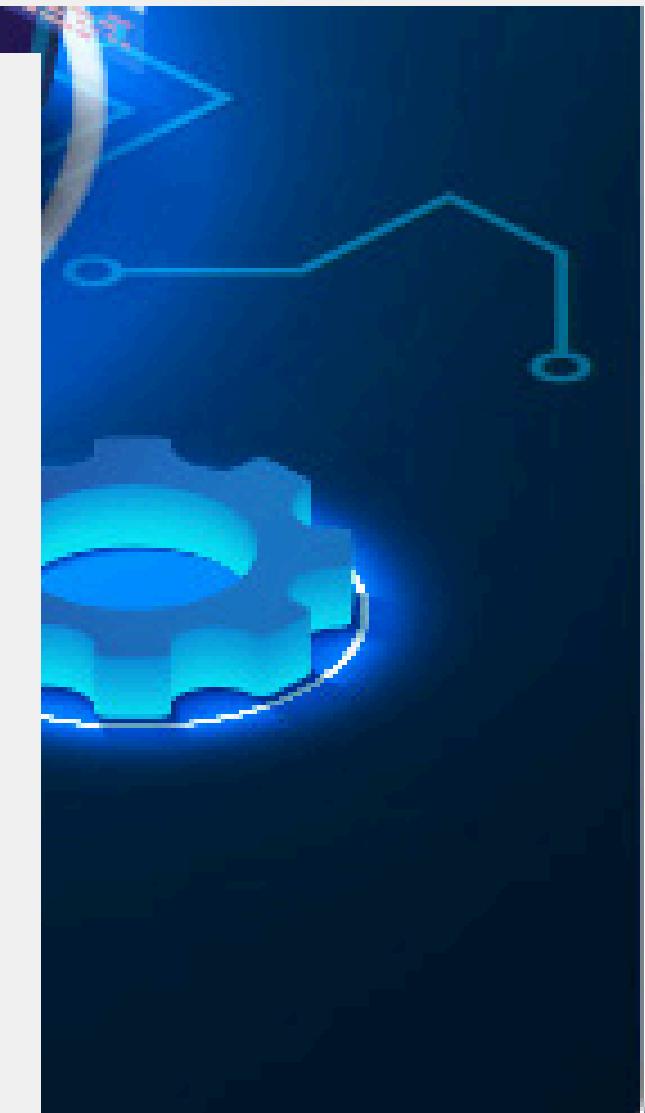
O núcleo é a camada mais básica do sistema operacional e é responsável por gerenciar recursos de hardware, como memória, CPU e dispositivos de entrada/saída. Ele fornece serviços para outras camadas do sistema operacional e garante que cada processo tenha acesso aos recursos necessários.





O gerenciador de processos é responsável por gerenciar os processos que estão em execução no sistema. Ele atribui recursos de CPU e memória para cada processo, controla a execução dos processos e garante que eles não interfiram uns com os outros. O gerenciador de processos também é responsável por lidar com erros e exceções que possam ocorrer durante a execução do processo.

O gerenciador de memória é responsável por gerenciar a memória do sistema, alocando e liberando espaço de memória conforme necessário. Ele também é responsável por proteger a memória do sistema contra processos mal-intencionados que possam tentar acessar áreas de memória que não lhes foram atribuídas.





O sistema de arquivos é responsável por gerenciar os arquivos armazenados no sistema. Ele fornece uma interface para que os usuários possam acessar, modificar e criar arquivos no sistema. O sistema de arquivos também é responsável por controlar o acesso aos arquivos, garantindo que apenas os usuários autorizados possam acessar e modificar arquivos específicos.



F U N D A M E N T O S
da
M A T E M Á T I C A
D I S C R E T A



Matemática discreta é um ramo da matemática que lida com estruturas matemáticas que são fundamentalmente discretas, em oposição às estruturas contínuas encontradas na matemática clássica. Ela é usada em uma variedade de áreas, incluindo ciência da computação, teoria da informação, criptografia, teoria dos grafos e muito mais.

Alguns dos fundamentos da matemática discreta incluem:

- 1. Lógica proposicional e de predicados:** Estuda as regras formais de inferência e as estruturas lógicas de proposições e predicados.
- 2. Conjuntos e teoria dos conjuntos:** Examina as propriedades e operações de conjuntos, incluindo união, interseção, diferença e produto cartesiano.
- 3. Teoria dos números:** Investigação dos números inteiros e suas propriedades, como divisibilidade, congruência, teorema fundamental da aritmética, e teoria dos números primos.
- 4. Indução matemática:** Um método de prova usado para estabelecer a verdade de uma afirmação matemática para todos os valores de uma variável inteira.
- 5. Combinatória:** Estuda a contagem e organização de objetos, incluindo permutações, combinações, princípio multiplicativo e princípio aditivo.
- 6. Teoria dos grafos:** Investigação de estruturas de grafos, que consistem em vértices (ou nós) conectados por arestas (ou linhas), incluindo propriedades de grafos, algoritmos de grafos e problemas relacionados.
- 7. Teoria dos conjuntos booleanos:** Estuda operações lógicas em conjuntos, como união, interseção e complemento, em um contexto booleano.
- 8. Álgebra booleana:** Trata de operações lógicas em valores binários (0 e 1), incluindo operações como AND, OR e NOT.

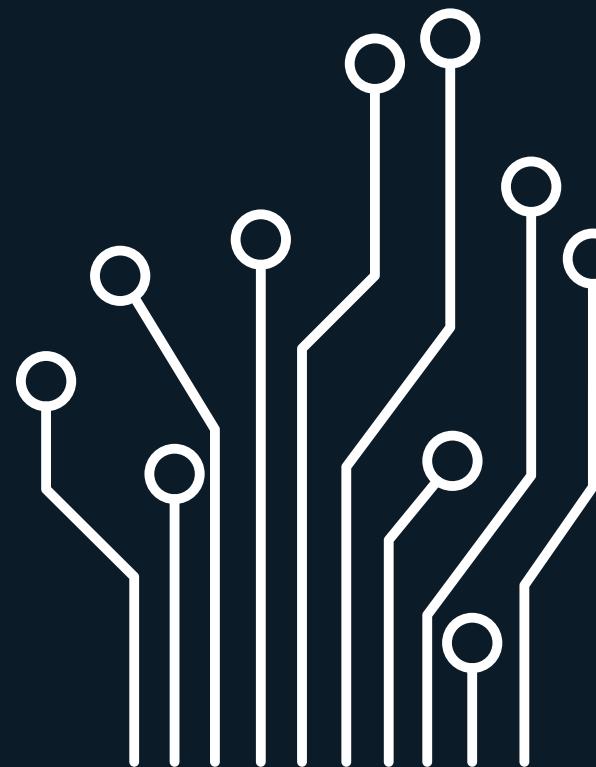
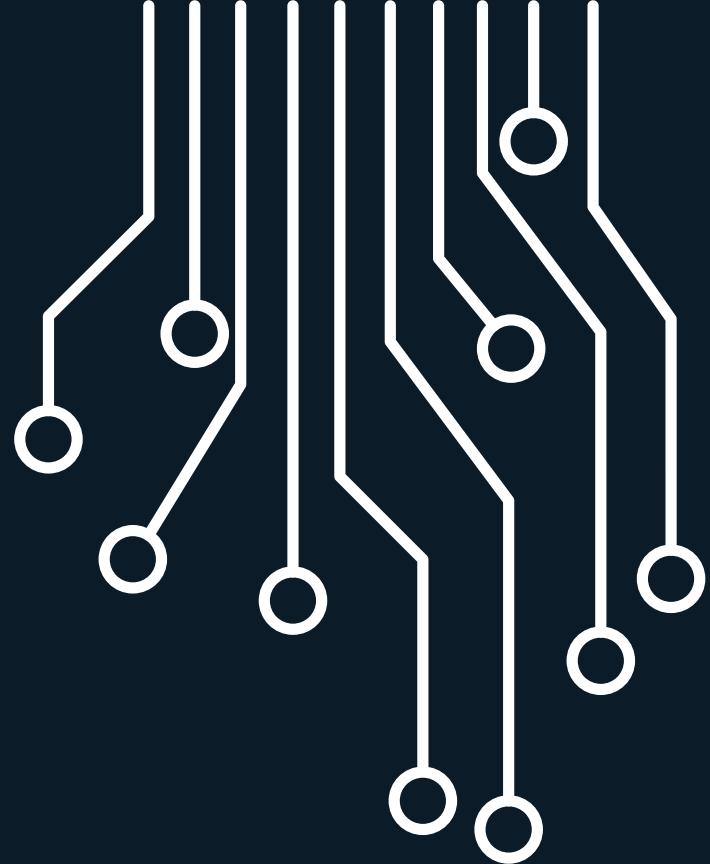
$$f(a, b, c) = \overline{a} \oplus \overline{b} + a \cdot \overline{b} + \overline{a} \cdot b \cdot c$$

ÁLGEBRA BOOLEANA

Boolean
Algebra

$$\begin{aligned}x \oplus y &= \overline{x} \cdot y + x \cdot \overline{y} \\x + y &= \overline{x} \cdot \overline{y} \\x \cdot y &= \overline{x} + \overline{y}\end{aligned}$$

A ÁLGEBRA BOOLEANA É UM RAMO DA MATEMÁTICA E DA LÓGICA QUE TRATA DE OPERAÇÕES LÓGICAS EM VARIÁVEIS BOOLEANAS. ESSAS VARIÁVEIS BOOLEANAS REPRESENTAM ESTADOS LÓGICOS QUE PODEM SER VERDADEIROS (1) OU FALSOS (0). ELA É FUNDAMENTAL PARA A CONCEPÇÃO E ANÁLISE DE CIRCUITOS DIGITAIS, SISTEMAS DE LÓGICA COMPUTACIONAL E ALGORITMOS BOOLEANOS EM CIÊNCIA DA COMPUTAÇÃO.



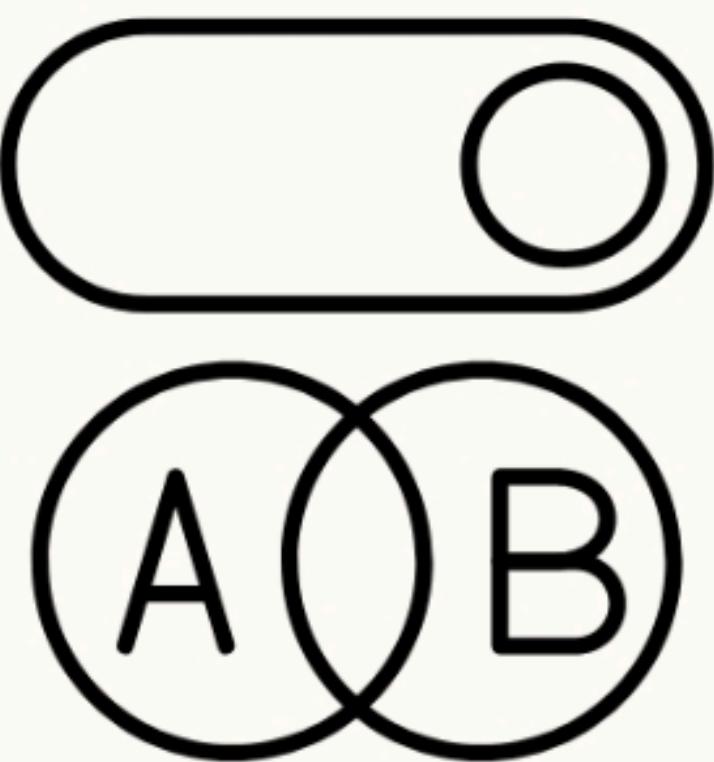
VARIÁVEIS BOOLEANAS

A álgebra booleana utiliza variáveis booleanas, que só podem assumir dois valores: verdadeiro (1) ou falso (0). Essas variáveis podem ser combinadas usando as operações lógicas AND, OR e NOT. A operação AND é verdadeira se ambas as variáveis são verdadeiras, enquanto a operação OR é verdadeira se pelo menos uma das variáveis é verdadeira. A operação NOT inverte o valor da variável.



Boolea Alge

$f(a, b, c)$



Tabelas-verdade

A álgebra booleana também pode ser representada graficamente através de tabelas-verdade, que mostram todas as combinações possíveis de valores para as variáveis e o resultado da operação lógica. Isso é útil para verificar se uma expressão lógica é verdadeira ou falsa.

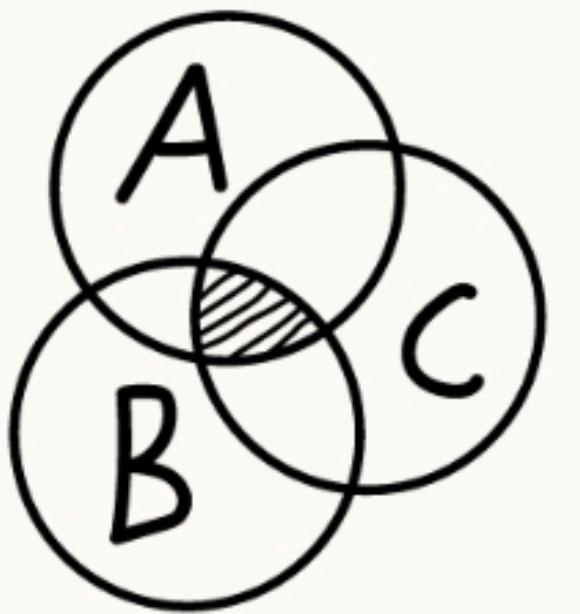
Álgebra booleana

Além disso, a álgebra booleana é usada para expressar e resolver problemas de lógica formal, como a resolução de provas matemáticas. Ele é também usado para representar proposições lógicas e para desenvolver teoremas em lógica matemática.

Inteligência artificial

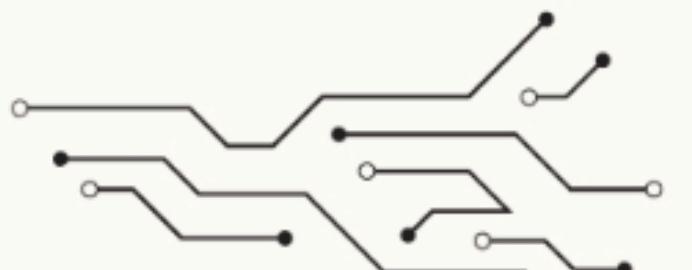
A álgebra booleana também é usada em inteligência artificial e aprendizado de máquina para representar conhecimento e para tomar decisões. Ela é usada para criar redes neurais e algoritmos de aprendizado de máquina.

Processamento de sinais digitais



Circuitos digitais

A álgebra booleana é amplamente utilizada em circuitos digitais, onde as operações lógicas são implementadas com circuitos eletrônicos. Os circuitos digitais são usados em dispositivos eletrônicos como computadores, telefones e televisores.



A álgebra booleana também é usada em processamento de sinais digitais, onde sinais são representados como sequências de bits e processados usando operações lógicas. Ela é usada para codificar, decodificar e comprimir sinais de áudio e vídeo.



Operações lógicas

Em resumo, a álgebra booleana é uma parte importante da matemática e da ciência da computação que estuda operações lógicas e sua aplicação em circuitos digitais.

$$\overline{b} + \overline{a} \cdot b \cdot c$$

$$y = \overline{x} \cdot y + x \cdot \overline{y}$$

$$\overline{x} + y = \overline{x} \cdot \overline{y}$$

$$\overline{x} \cdot y = \overline{x} + \overline{y}$$

Álgebra Booleana

Álgebra Booleana é o ramo da Matemática dedicado ao estudo de operações lógicas binárias, também conhecidas como operações de álgebra Booleana. Estas operações incluem a conjunção (AND), a disjunção (OR) e a negação (NOT). A Álgebra Booleana foi inventada pelo matemático inglês George Boole em 1854.

$\bar{a} \cdot b \cdot c$

$\bar{x} \cdot v + x \cdot \bar{v}$

É importante entender que a Álgebra Booleana é baseada em operações lógicas binárias. Em outras palavras, ela trabalha com apenas dois valores: verdadeiro (1) e falso (0). Estes valores são usados para representar as relações lógicas entre duas ou mais expressões.

Boole
Álge

A Álgebra Booleana usa símbolos para expressar as operações lógicas binárias. Os principais símbolos usados são a conjunção (AND), a disjunção (OR) e a negação (NOT). Estes símbolos permitem que os matemáticos representem relações lógicas entre expressões de forma simples e intuitiva.

Boole
Alg

A Álgebra Booleana também é usada para representar funções lógicas. Uma função lógica é uma relação entre duas ou mais variáveis, onde cada variável tem apenas dois valores possíveis: verdadeiro ou falso. Estas funções são usadas para modelar problemas lógicos e são comumente usadas para representar circuitos digitais.

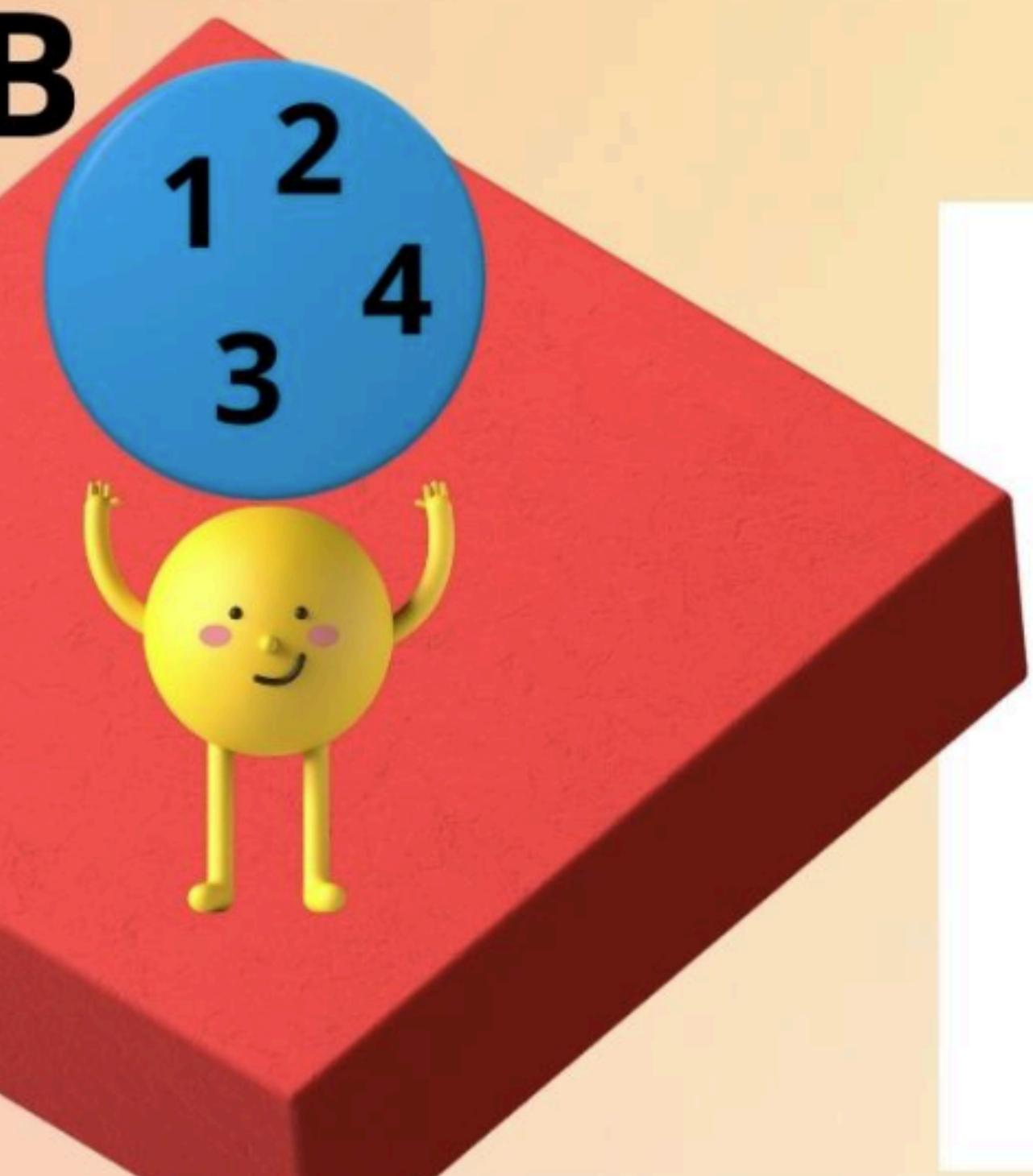
O estudo da Álgebra Booleana também inclui a análise de tabelas de verdade. Uma tabela de verdade é um diagrama que representa uma função lógica. Ela contém todos os possíveis valores que a função pode assumir para cada possível combinação de entradas.

Boolean Algebra

A Álgebra Booleana também é usada para simplificar sistemas lógicos complexos. Estes sistemas podem ser simplificados usando técnicas como a regra de simplificação de Karnaugh, a regra de simplificação de Quine-McCluskey ou a técnica de tabela de verdade.

$\bar{a} \cdot b \cdot c$

Conjuntos



Coleção de objetos

Na matemática, um conjunto é uma coleção de objetos matemáticos, geralmente denotados por letras maiúsculas, como A, B, C, etc. Esses objetos podem ser números, letras, palavras, figuras geométricas, funções, entre outros.

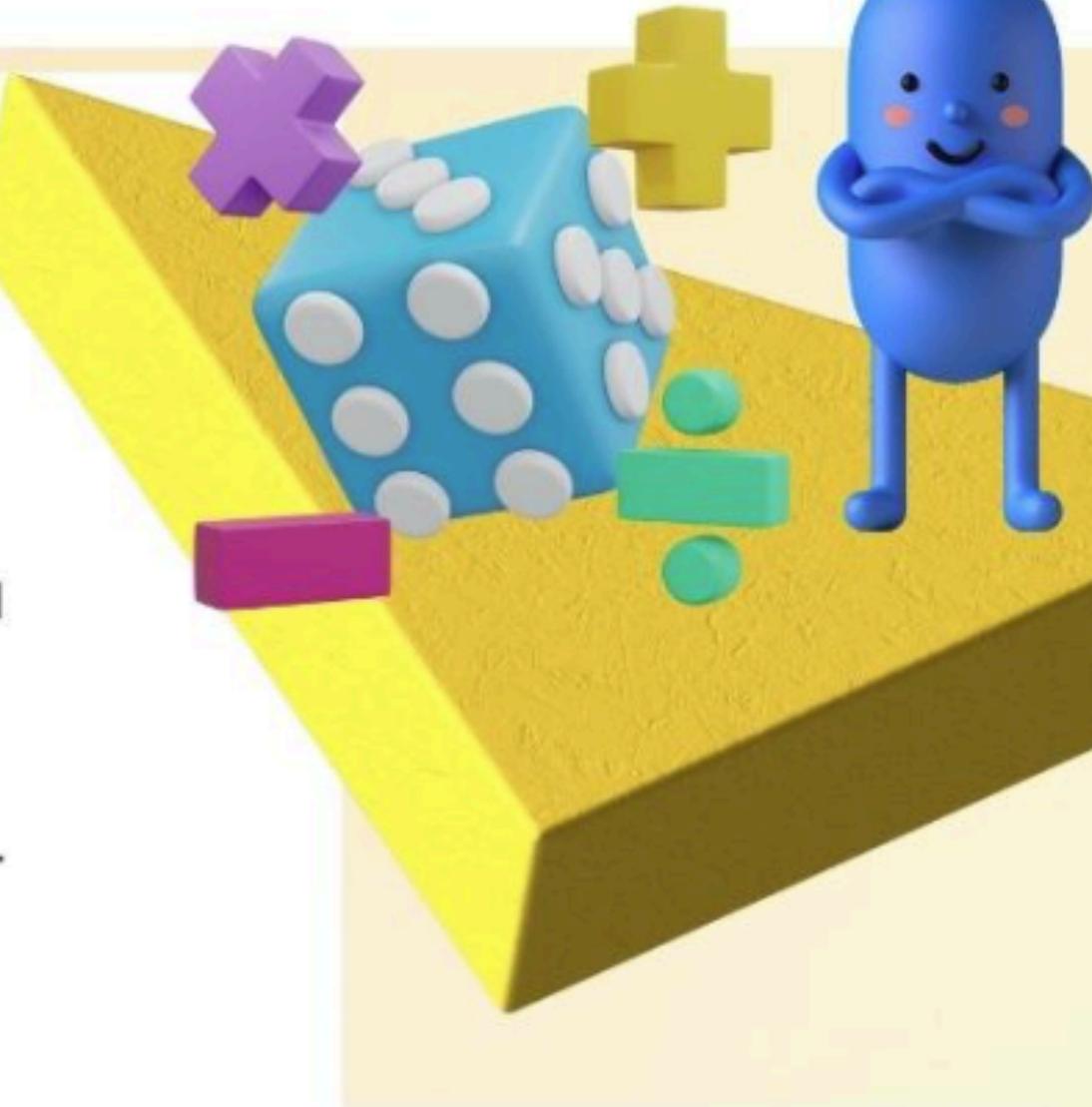


A



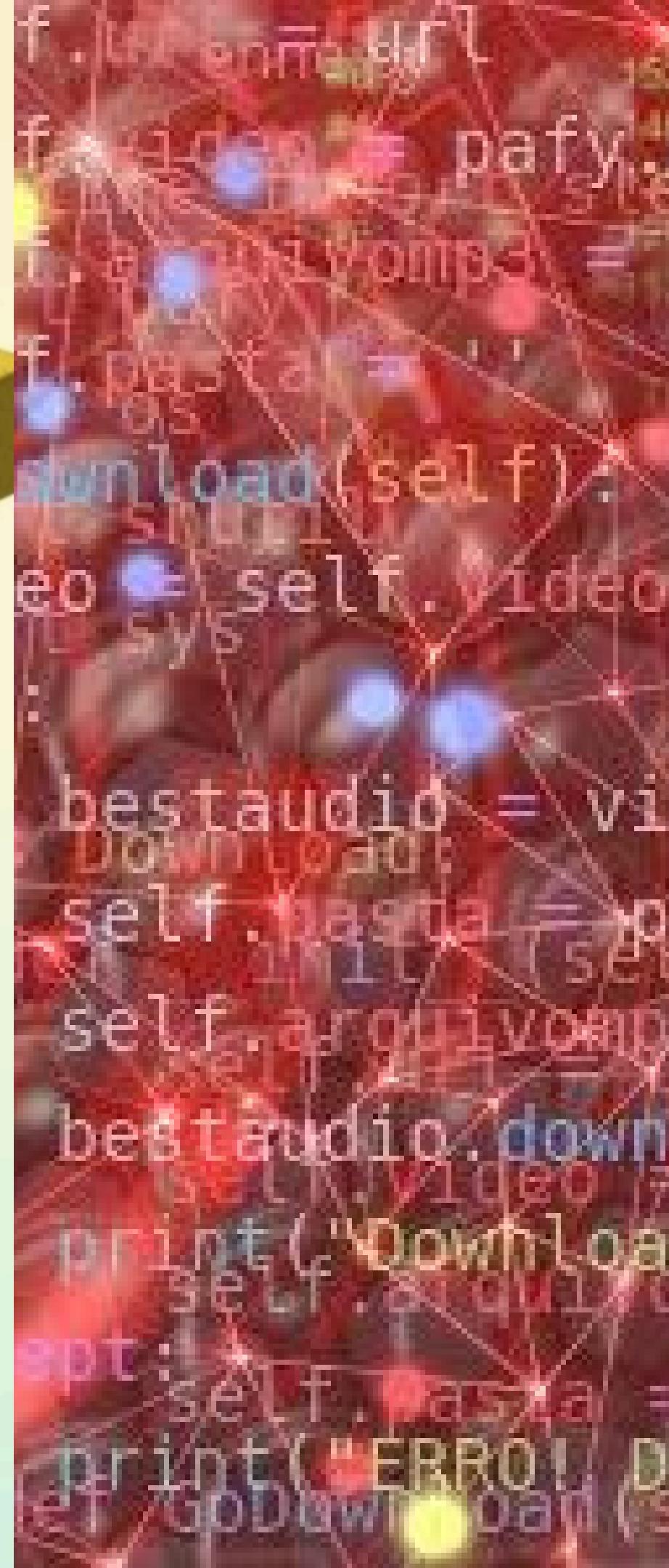
Finito ou infinito

Os conjuntos podem ser finitos ou infinitos e podem ser representados de diferentes maneiras, como por extensão, por compreensão, por diagrama de Venn, entre outras.



Elementos

Os elementos de um conjunto são únicos, ou seja, não há elementos repetidos em um conjunto. Além disso, os conjuntos não têm ordem, ou seja, não importa a ordem em que os elementos são listados.



Operações

As operações mais comuns em conjuntos são união, interseção, diferença e complemento. A união de dois conjuntos A e B é o conjunto que contém todos os elementos que pertencem a A ou a B. A interseção de A e B é o conjunto que contém todos os elementos que pertencem a A e a B. A diferença de A e B é o conjunto que contém todos os elementos que pertencem a A, mas não pertencem a B. O complemento de A em relação a um conjunto universo U é o conjunto de todos os elementos de U que não pertencem a A.





Os conjuntos têm muitas aplicações em matemática e em outras áreas, como ciência da computação, estatística, teoria dos conjuntos, teoria dos grafos, entre outras.

SISTEMAS DE NUMERAÇÃO

Os sistemas de numeração são métodos utilizados para representar números de forma sistemática, utilizando símbolos ou algarismos. Existem diferentes sistemas de numeração, cada um com suas próprias características e aplicações. Alguns dos sistemas de numeração mais comuns são:

Sistema Decimal: O sistema decimal é o mais utilizado no dia a dia. Ele é baseado em 10 símbolos (0, 1, 2, 3, 4, 5, 6, 7, 8 e 9) e usa posições para representar valores. Cada posição tem um valor que é uma potência de 10, começando da direita para a esquerda, com valores crescentes (1, 10, 100, 1000, etc.).

Por exemplo, o número 345 no sistema decimal é interpretado como $3 \times 10^2 + 4 \times 10^1 + 5 \times 10^0$, ou seja, $300 + 40 + 5 = 345$.

Sistema Binário: O sistema binário é amplamente utilizado em computação e eletrônica digital. Ele é baseado em apenas dois símbolos (0 e 1) e cada posição representa uma potência de 2.

Por exemplo, o número 101 no sistema binário é interpretado como $1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$, ou seja, $4 + 0 + 1 = 5$.

Sistema Octal: O sistema octal é baseado em 8 símbolos (0, 1, 2, 3, 4, 5, 6 e 7) e é utilizado principalmente em programação de computadores. Cada posição representa uma potência de 8.

Sistema Hexadecimal: O sistema hexadecimal é amplamente utilizado em computação e eletrônica digital. Ele é baseado em 16 símbolos, representados por 0-9 e as letras A-F (ou a-f) para os valores de 10 a 15.

Cada posição representa uma potência de 16.

Por exemplo, o número 2A7 no sistema hexadecimal é interpretado como $2 \times 16^2 + 10 \times 16^1 + 7 \times 16^0$, ou seja, $512 + 160 + 7 = 679$.

Além desses sistemas, existem outros menos comuns, como o sistema de numeração romana (utilizado historicamente em inscrições e documentos antigos) e sistemas de numeração mais exóticos usados em contextos específicos.

Cada sistema de numeração tem suas próprias vantagens e desvantagens em relação à representação, manipulação e interpretação de números, e a escolha do sistema depende do contexto e dos requisitos específicos de uma aplicação.



LÓGICA DE PROGRAMAÇÃO

Conceitos de Interpretadores e de Compiladores

INTERPRETADORES E COMPILADORES

Interpretadores e compiladores são ferramentas usadas na programação para transformar o código fonte de um programa em instruções que o computador possa entender e executar.

INTERPRETADORES

Um interpretador é um programa que lê e executa diretamente o código fonte de um programa. Ele lê uma linha de cada vez e executa imediatamente, sem criar um arquivo executável separado. Isso torna a execução mais lenta, mas mais fácil para o desenvolvedor, pois permite testar o código em tempo real.

COMPILADOR

Já um compilador é um programa que traduz o código fonte em um arquivo executável. O compilador lê todo o código de uma só vez e verifica erros antes de criar o arquivo executável. A execução é mais rápida, mas o processo de compilação é mais lento e exige mais recursos.

VANTAGENS E DESVANTAGENS

interpretadores executam o código fonte diretamente, enquanto compiladores traduzem o código fonte para um arquivo executável. Ambos têm suas vantagens e desvantagens e são usados em diferentes situações, dependendo do tipo de projeto e das necessidades do desenvolvedor.

PYTHON BÁSICO



python™

```
17     string sInput;
18     int iLength, iN;
19     double dblTemp;
20     bool again = true;
21
22     while (again) {
23         iN = -1;
24         again = false;
25         getline(cin, sInput);
26         system("cls");
27         stringstream(sInput) >> dblTemp;
28         iLength = sInput.length();
29         if (iLength < 4) {
30             if (again = true);
31             continue;
32         } else if (sInput[iLength - 3] != '.') {
33             again = true;
34             continue;
35         } while (++iN < iLength) {
36             if (isdigit(sInput[iN])) {
37                 continue;
38             } else if (iN == (iLength - 3)) {
39                 again = true;
40             }
41         }
42     }
43 }
```

Python é uma linguagem de programação de alta nível, interpretada e de código aberto. Foi criada pelo holandês Guido Van Rossum em 1991 e tem sido amplamente utilizada em muitas áreas incluindo ciência de dados, inteligência artificial, desenvolvimento web e automação de tarefas. Python é conhecido por sua sintaxe simples e legível, o que o torna fácil de aprender e usar.



Python foi projetado com ênfase na legalidade e simplicidade foi concebido como uma linguagem de programação de propósito geral. Com uma sintaxe limpa e uma filosofia que enfatiza a importância da clareza, facilidade de uso da comunidade.

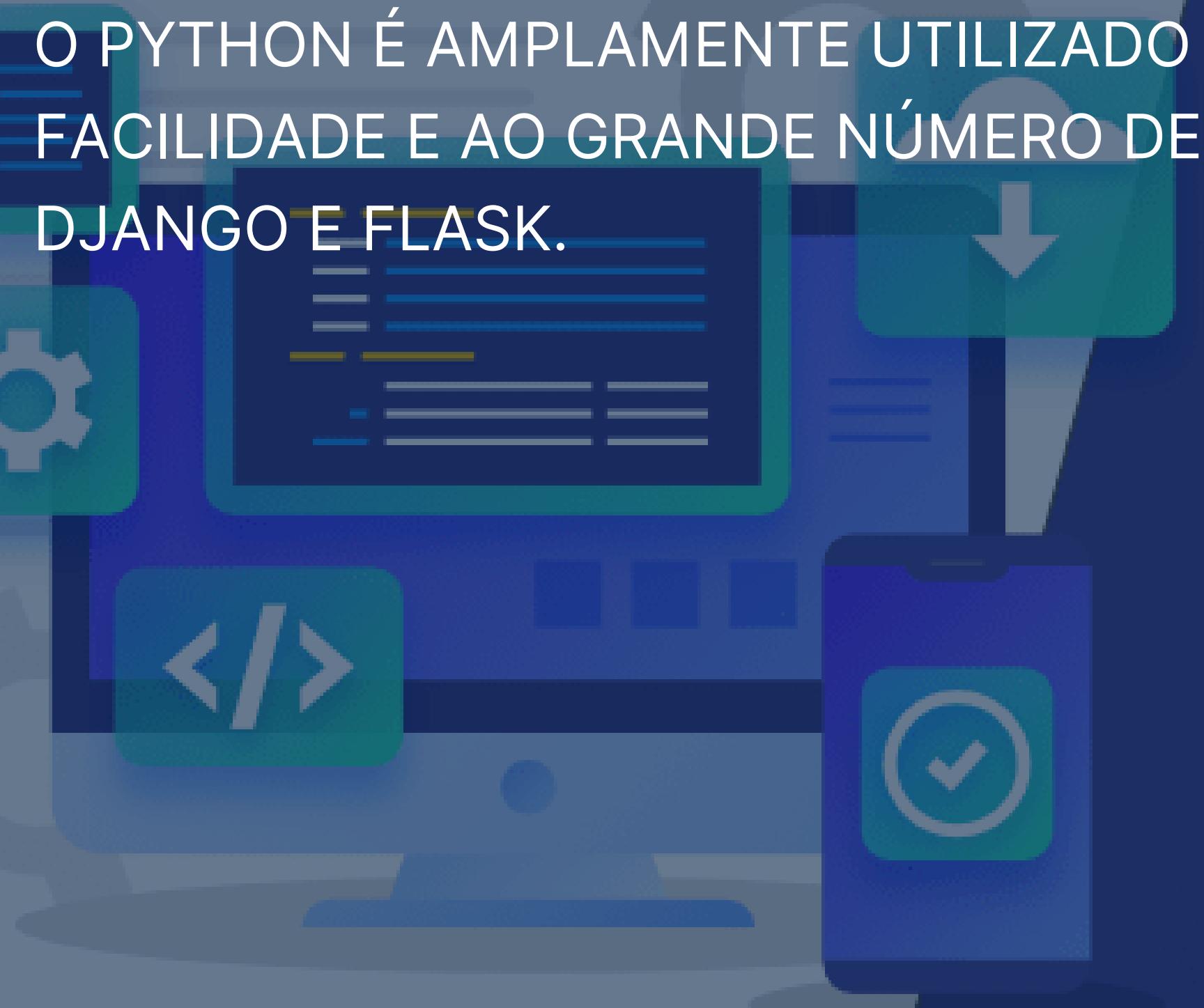
Python possui uma grande variedade de ferramentas de desenvolvimentos como IDEs e editores de texto, que ajudam a facilitar o processo de desenvolvimento.

Python

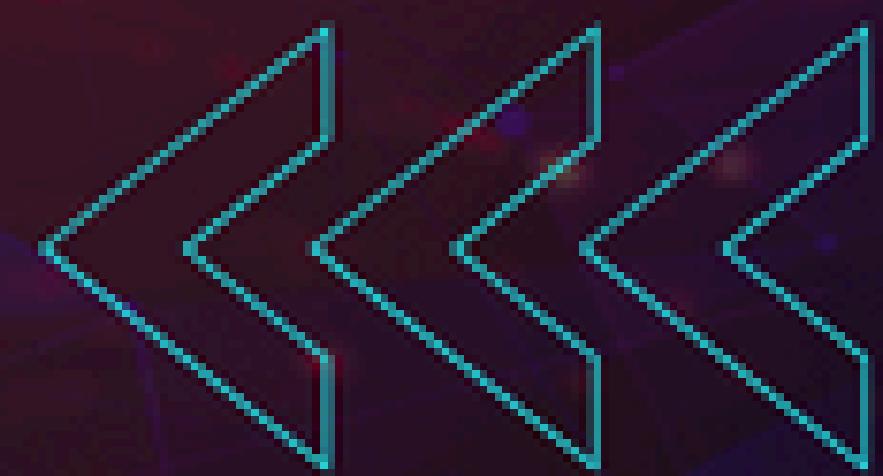


O PYTHON TEM UMA DAS SUAS PRINCIPAIS VANTAGENS A SUA GRANDE COMUNIDADE E BIBLIOTECA DE MÓDULOS, CHAMADAS DE PYPI (PYTHON PACKAGE INDEX).

O PYTHON É AMPLAMENTE UTILIZADO EM DESENVOLVIMENTO WEB DEVIDO À SUA FACILIDADE E AO GRANDE NÚMERO DE FRAMEWORKS WEB DISPONÍVEIS, COMO DJANGO E FLASK.



IDES





IDEs é uma ferramenta de software que oferece um conjunto abrangente de recursos para facilitar o desenvolvimento de software. Em essência uma IDE é um ambiente unificado onde os desenvolvedores podem escrever, compilar, depurar e testar seus códigos de maneira eficiente.

As Principais IDEs são:

- . Pycharm
- . Visual Studio Code (VScode)
- . Spyder
- . Jupyter
- . IDLE



Tipos Primitivos, Constantes e Variáveis em Python



OS TIPOS PRIMITIVOS

Em Python, os tipos primitivos se referem aos tipos básicos de dados que a linguagem suporta, como números inteiros, números de ponto flutuante, booleanos e strings. Cada um desses tipos de dados é usado para armazenar um tipo específico de informação e pode ser utilizado em diferentes operações.



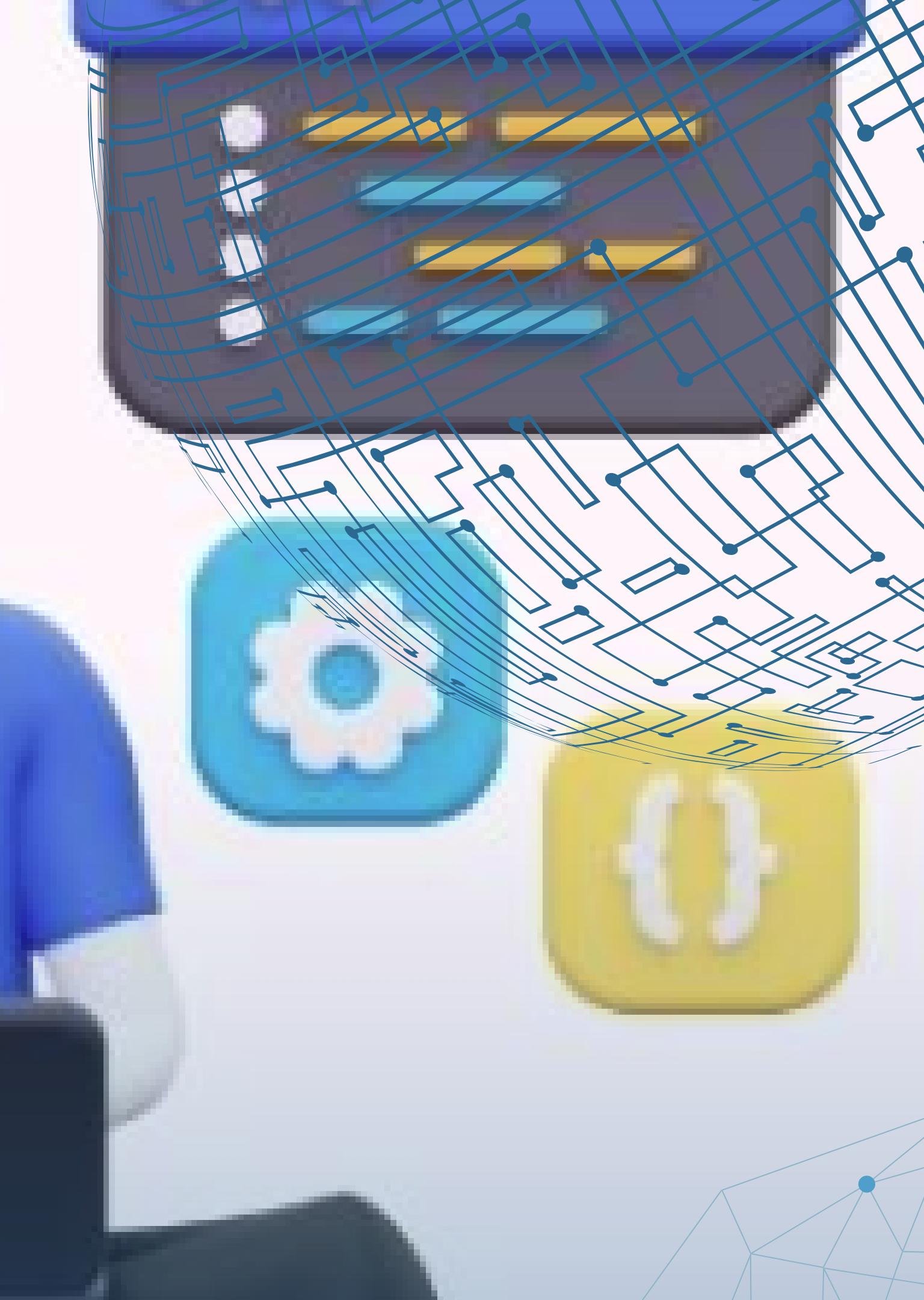
CONTANTES

As constantes são valores que nunca mudam, ou seja, são fixos. Em Python, uma constante pode ser definida como uma variável que é atribuída a um valor fixo e nunca é alterada durante a execução do programa. Essas constantes podem ser usadas em operações aritméticas, comparações, entre outros.



VARIÁVEIS

As variáveis em Python são usadas para armazenar valores que podem ser alterados ao longo do tempo durante a execução do programa. Uma variável é definida por um nome e um valor, e seu valor pode ser atualizado conforme necessário durante a execução do programa.



VARIÁVEIS EM PYTHON

Em Python, as variáveis não precisam ser explicitamente declaradas antes de serem usadas, como ocorre em outras linguagens. Quando uma variável é usada pela primeira vez em um programa, ela é criada automaticamente pelo interpretador.

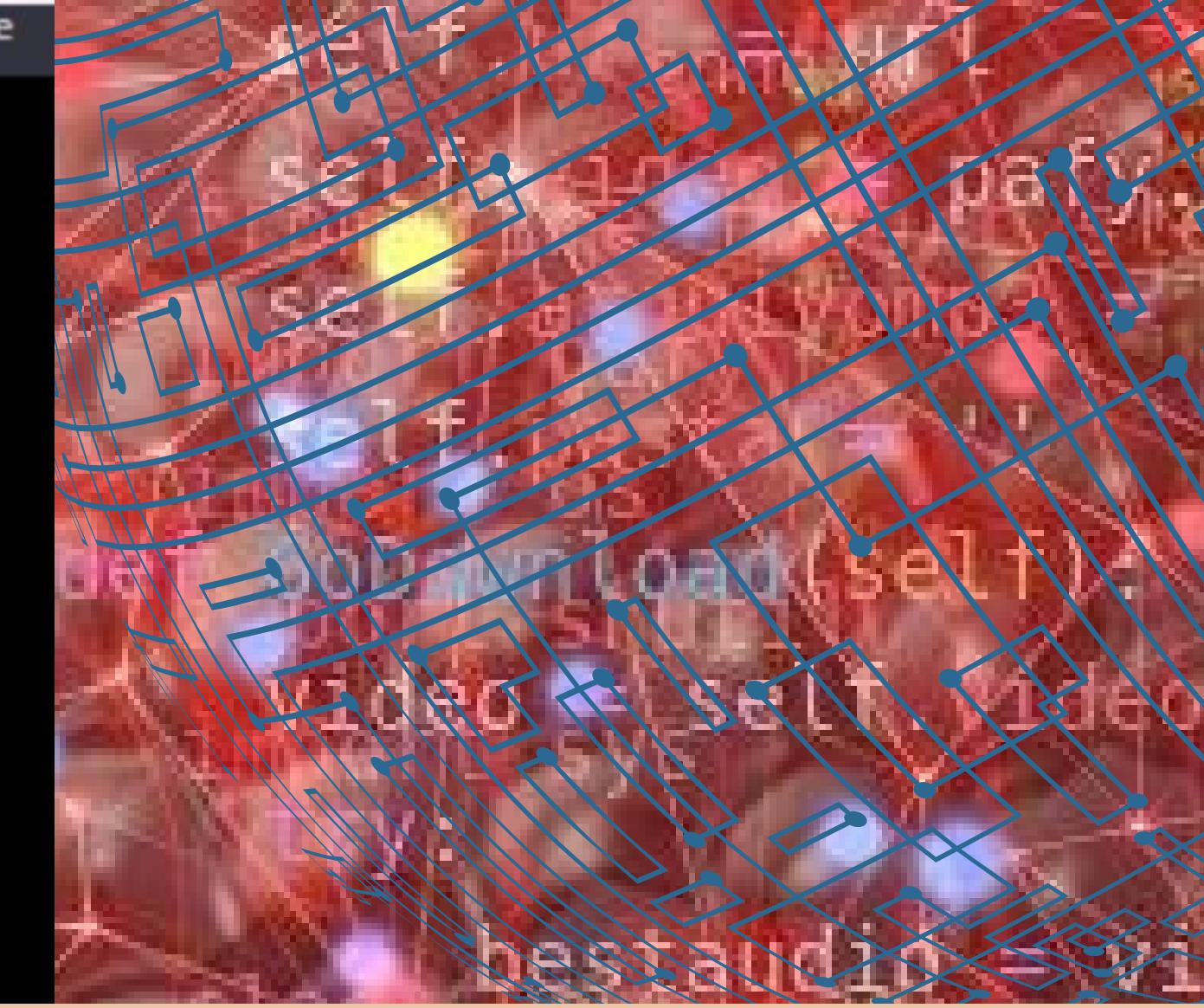
TIPOS PRIMITIVOS

Os tipos primitivos, constantes e variáveis são usados em Python para armazenar diferentes tipos de dados e valores. O interpretador Python pode identificar automaticamente o tipo de dado que está sendo utilizado em uma variável, dependendo do valor atribuído a ela.

NÚMEROS INTEIROS

Em Python, números inteiros podem ser representados por valores numéricos inteiros, ou inteiros, que não possuem casas decimais. Por exemplo, o número inteiro 10 pode ser representado em Python simplesmente escrevendo "10". O mesmo pode ser feito com números inteiros negativos, como -10.

Aqui está um exemplo simples de como trabalhar com números inteiros em Python:



Copy code

```
# atribuindo o valor inteiro 10 à variável "a"  
a = 10  
  
# atribuindo o valor inteiro -5 à variável "b"  
b = -5  
  
# somando as variáveis "a" e "b"  
c = a + b  
  
# imprimindo o valor da variável "c"  
print(c)
```

INTEIROS

Neste exemplo, a variável "a" é definida como o número inteiro 10, enquanto a variável "b" é definida como o número inteiro negativo -5. Em seguida, essas duas variáveis são adicionadas e o resultado é armazenado na variável "c". Por fim, o valor da variável "c" é impresso na tela.



PYTHON INTEIROS

Note que não é necessário declarar explicitamente o tipo de dados como "inteiro" em Python. A linguagem é capaz de inferir o tipo de dados a partir do valor atribuído a uma variável.

Python operadores



Python suporta uma variedade de operadores que podem ser usados para realizar operações em valores e variáveis. Os principais operadores em Python são:

Operadores aritméticos

Operadores Aritméticos: São usados para realizar operações matemáticas básicas, como adição (+), subtração (-), multiplicação (*), divisão (/), módulo (%), exponenciação (**).

**

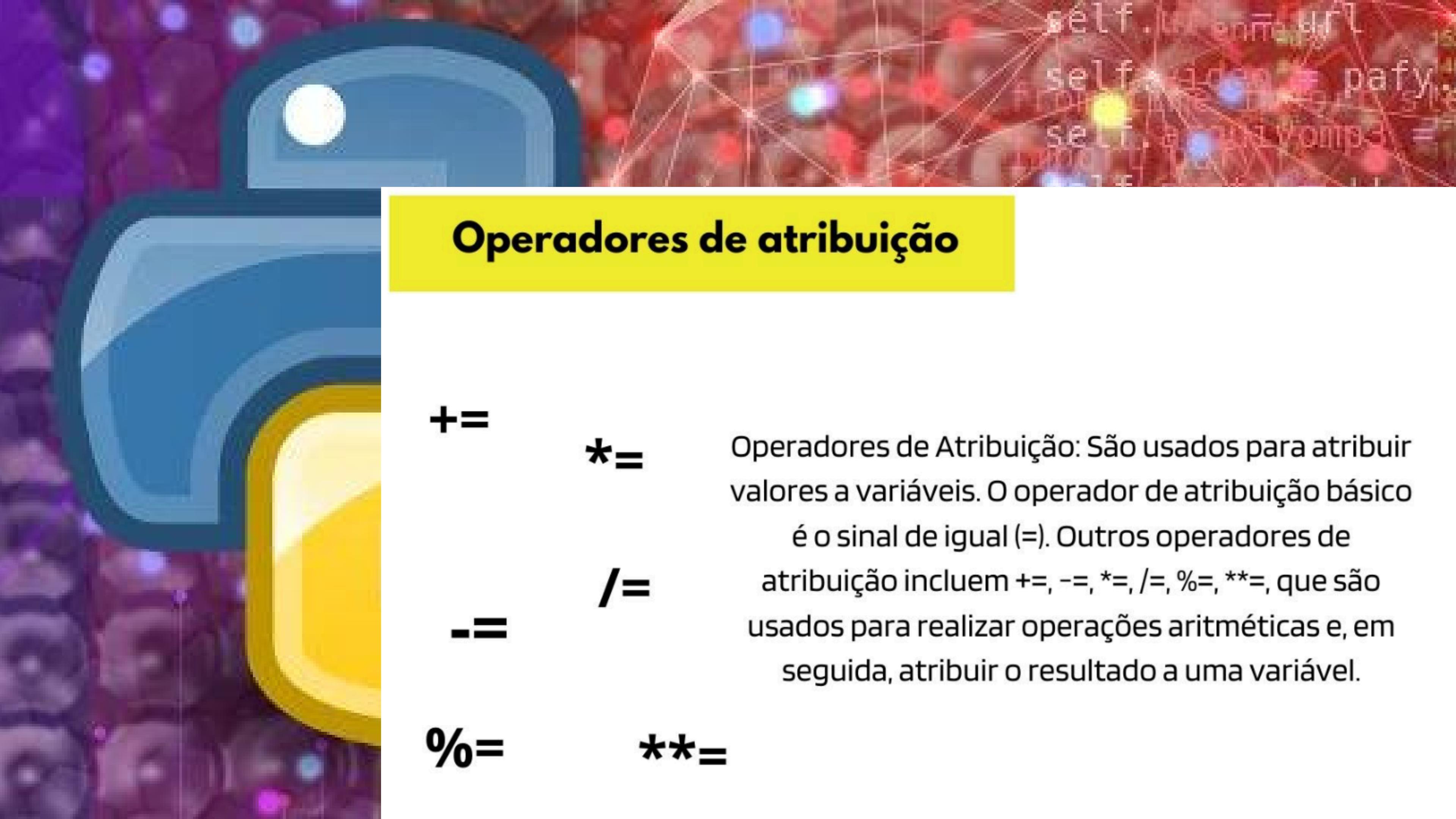
+

*

/

%





Operadores de atribuição

$+=$

$*=$

$-=$

$/=$

$%=$

$=$**

Operadores de Atribuição: São usados para atribuir valores a variáveis. O operador de atribuição básico é o sinal de igual ($=$). Outros operadores de atribuição incluem $+=$, $-=$, $*=$, $/=$, $%=$, $**=$, que são usados para realizar operações aritméticas e, em seguida, atribuir o resultado a uma variável.

Operadores comparação

> <

== !=

>= <=

Operadores de Comparação: São usados para comparar dois valores ou variáveis e produzir um resultado booleano (True ou False). Os operadores de comparação incluem == (igual a), != (diferente de), > (maior que), < (menor que), >= (maior ou igual a), <= (menor ou igual a).

