**Sobre Linguagens de Programação e Seus Interpretadores:** As linguagens de programação, devido à sua complexidade, requerem interpretadores que compilam o código em linguagem de máquina.

**Linguagens Compiladas:**

* Após erros, é necessário recompilar e executar o código.
* Transformam código em linguagem de máquina (binário).
* Oferecem maior rapidez e eficiência.
* Requerem compilação ("build").
* Exemplos incluem C, C++ e Rust.

**Linguagens Interpretadas:**

* Processadas linha a linha.
* Geralmente mais lentas que as compiladas.
* Dispensam o processo de compilação.
* Usam JIT (Just-In-Time) para compilar e interpretar dinamicamente.
* Exemplos incluem JavaScript, PHP e Python.

**Linguagem C:**

* De nível intermediário, permite manipulação de estruturas de controle e acesso a recursos de baixo nível, como a memória.

**Node.js:**

* Permite a execução de JavaScript no servidor de forma independente, sem navegador.

**Definição e Construção de Algoritmos:**

* Um algoritmo é uma sequência definida de passos para completar uma tarefa.
* A construção eficaz de um algoritmo envolve entender claramente o problema, identificar entradas e saídas, e determinar os processos necessários para as transformações desejadas.

**Variáveis na Computação:**

* Armazenadas na RAM com um endereço específico para sua localização.