

## Capítulo 2 – Do Algoritmo ao Pseudocódigo (Introdução ao Visualg)

**Carga horária:** 4h (2h teoria + 2h prática)

**Etapas:** Fundamentos de Representação Algorítmica

**Objetivo:** Compreender como as instruções lógicas que o ser humano cria mentalmente podem ser representadas em uma linguagem compreensível para o computador. Aprender a estruturar algoritmos utilizando pseudocódigo e introduzir o uso do Visualg como ferramenta prática de aprendizagem.

### 2.1. Do pensamento lógico à escrita do algoritmo

Antes de programar, é preciso **pensar como o computador**, de forma estruturada, sem pular etapas.

O programador precisa descrever passo a passo **o que deve ser feito, em que ordem e com que finalidade**.

Esse raciocínio é o mesmo que usamos em nosso dia a dia: seguir uma receita, montar um móvel ou resolver um problema. A diferença é que o computador **não interpreta intenções**, ele **segue instruções exatas**.

Por isso, o primeiro passo é **traduzir a lógica humana para uma linguagem clara e sem ambiguidades**.

Essa tradução é o que chamamos de **algoritmo**.

### 2.2. O que é um Pseudocódigo?

O pseudocódigo é uma forma de **representar o algoritmo por meio de palavras e comandos lógicos**, escritos em linguagem natural estruturada.

É o meio-termo entre o **fluxograma visual** e o **código de programação real**.

A ideia é que o você escreva suas instruções **em português estruturado**, respeitando uma **ordem lógica**.

Assim, o computador ainda não executa, mas **o raciocínio já é o mesmo** que será usado nas linguagens formais.

### Exemplo comparativo

Tipo de representação	Exemplo
Linguagem natural	“Fazer café: ferver a água, colocar o pó, coar e servir.”
Pseudocódigo	Início → Ferver água → Colocar pó → Coar → Servir → Fim

O pseudocódigo elimina interpretações ambíguas e **impõe disciplina lógica**, o que transforma ideias em instruções precisas.

### 2.3. Estrutura básica de um algoritmo

Um algoritmo segue sempre uma estrutura bem definida. É como a espinha dorsal do raciocínio lógico, e toda linguagem de programação mantém esse formato.

Algoritmo "nome"

Início

[comandos]

FimAlgoritmo

Essa estrutura é **a base de qualquer programa**, seja ele simples ou complexo.

#### Entendendo cada parte

- **Algoritmo "nome"** - dá identidade ao código. Serve como título.
- **Início** - indica onde a execução começa.
- **Comandos** - são as ações executadas, uma por linha.
- **FimAlgoritmo** - encerra a execução. Nada vem depois dele.

Essa sequência cria um **fluxo controlado de pensamento**, semelhante a uma linha de montagem: o computador **entra, processa e sai**.

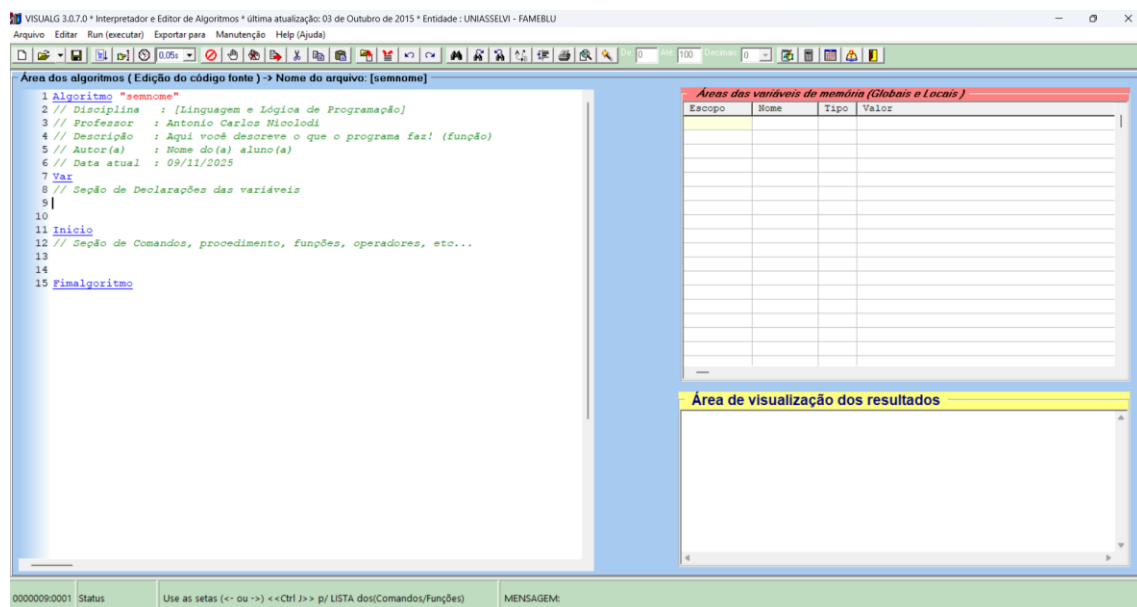
## 2.4. O ambiente Visualg

O Visualg é uma ferramenta que simula a execução de algoritmos escritos em português estruturado.

Ele foi criado para **ensinar lógica de programação sem exigir conhecimento prévio de linguagens complexas.**

## Elementos principais do Visualg

1. **Área de edição** - onde se escreve o algoritmo.
2. **Janela de saída** - mostra o resultado da execução.
3. **Verificador automático** - identifica erros de escrita ou estrutura.
4. **Comando de execução (F9)** - inicia a simulação.



## Por que usar o Visualg?

- Ele **permite errar e aprender** rapidamente.

- Mostra como o raciocínio lógico **vira comportamento do computador**.
- Serve como um “laboratório de lógica”, ideal para quem está começando.

💡 *Dica:* Observe **o que o programa faz**, mas também **como ele faz**. Cada execução é uma oportunidade de compreender o “pensar lógico”.

## 2.5. Saídas de dados – Fazendo o computador falar

Para que o computador mostre algo na tela, utilizamos o comando *Escreva()*.

Esse comando é a forma mais simples de **comunicação entre o programa e o usuário**.

### Estrutura

```
Escreva("mensagem")
```

Tudo o que estiver entre aspas será exibido exatamente como digitado.

### Exemplo – O primeiro algoritmo

Algoritmo "primeiro"

Início

```
Escreva("Bem-vindo ao Visualg!")
```

```
Escreva("Agora o computador entende suas instruções.")
```

FimAlgoritmo

Ao executar, o computador apresentará:

*Bem-vindo ao Visualg!*

*Agora o computador entende suas instruções.*

#### O que aprendemos aqui:

- Cada linha representa uma **ação independente**.
- A execução é **sequencial**, da primeira à última instrução.
- O computador não “pula” nem “interpreta”, ele apenas **segue ordens**.

### 2.6. Pensando o algoritmo antes de escrever

Antes de abrir o Visualg, o você precisa **visualizar a lógica**.

Essa etapa é crucial: pensar o que deve acontecer, em que ordem e com que resultado.

#### Exemplo de raciocínio:

Quero que o programa cumprimente o usuário e, em seguida, diga que está iniciando uma tarefa.

Planejamento no papel:

1. Mostrar mensagem de boas-vindas.
2. Mostrar mensagem de início da tarefa.

Depois de pensar, basta traduzir em pseudocódigo:

Algoritmo "início"

Início

Escreva("Olá, seja bem-vindo!")

Escreva("O sistema está iniciando suas funções...")

FimAlgoritmo

💡 Assim o passamos a perceber que **pensar vem antes de digitar**.

A lógica deve nascer na cabeça, não no teclado.

## 2.7. Explorando novas possibilidades

A partir de agora, poderemos **criar nossos próprios textos**, combinando frases, linhas e mensagens.

### Experimente

Algoritmo "mensagens"

Início

`Escreva("Este é o meu primeiro programa!")`

`Escreva("-----")`

`Escreva("Aprender lógica é aprender a pensar!")`

FimAlgoritmo

O comando `Escreva()` pode ser substituído por **`EscrevaL()`** (com “L”) para pular uma linha após cada mensagem.

Isso torna a saída mais organizada e agradável de ler.

💡 Essa simples diferença ajuda e faz perceber **a importância da formatação**, algo essencial em programação profissional.

## 2.8. Prática guiada – Criando uma apresentação pessoal

Agora que você já compreende a estrutura e o funcionamento do `Escreva()`, é hora de aplicar a lógica em um contexto real.

### Desafio:

Crie um algoritmo que mostre na tela:

- Seu nome;
- O curso que está realizando;
- Um motivo pelo qual deseja aprender programação.

Exemplo:

Algoritmo "apresentacao"

Início

`EscrevaL("Meu nome é Hilton.")`

`EscrevaL("Sou aluno do curso Técnico em Desenvolvimento de Sistemas.")`

`EscrevaL("Aprendo programação para transformar ideias em soluções reais.")`

FimAlgoritmo

O resultado é uma mensagem organizada e coerente, um retrato lógico do aluno dentro do ambiente do Visualg.

## 2.9. Conclusão

O pseudocódigo é o **primeiro idioma entre você e o computador**.

Ele ensina a **pensar de forma estruturada**, a **organizar ideias**, e a **comunicar intenções de maneira precisa**.

“Programar é ensinar a máquina a pensar como você, mas para isso, você precisa aprender a pensar como ela.” - *Hilton Elias Tomaszewski dos Santos, 2025*.

**Ao final deste capítulo, você deverá ser capaz de:**

- Compreender a estrutura básica de um algoritmo;
- Traduzir raciocínios simples em pseudocódigo;
- Usar corretamente o comando `Escreva()` e `EscrevaL()`;
- Entender a execução sequencial no Visualg;
- Criar algoritmos com clareza e formatação organizada.