Manual de Programação no CompilAI

Table of Contents

# Manual de Programação no CompilAI

Versão: 1.0  
Data: 25/09/2025  
Autor: Equipe de Documentação

## Objetivo

Este manual ensina, passo a passo, como programar no CompilAI, cobrindo desde a estrutura básica de um programa até estruturas de seleção, repetição, organização de cabeçalhos, funções e procedimentos, vetores e matrizes, e exemplos completos de algoritmos clássicos.

Observação: a sintaxe utilizada é inspirada em notações educacionais de pseudo‑código (Portugol). Palavras‑chave podem variar na sua instalação do CompilAI; adapte os identificadores se sua distribuição usar nomes diferentes.

## Sumário

* Introdução e primeiro programa
* Estrutura do arquivo e cabeçalho
* Tipos, variáveis e constantes
* Entrada e saída
* Operadores e expressões
* Estruturas de seleção (se/senao, escolha/caso)
* Estruturas de repetição (para, enquanto, repita‑até)
* Funções e procedimentos
* Vetores e matrizes
* Strings e utilitários comuns
* Boas práticas
* Exemplos completos
* Exercícios propostos
* Apêndice A: Tabela de palavras‑chave
* Apêndice B: Modelos prontos (cabeçalho, função, laços)

## Introdução e primeiro programa

Seu primeiro programa costuma imprimir uma mensagem. No CompilAI, a estrutura geral segue bloco algoritmo … inicio … fim\_algoritmo.

algoritmo "OlaMundo"  
 inicio  
 escreva("Olá, mundo!\n")  
 fim\_algoritmo

Execute para ver a mensagem no console. Se estiver usando um IDE do CompilAI, clique em Executar. Se usar linha de comando, consulte a documentação da sua distribuição.

## Estrutura do arquivo e cabeçalho

Organize cada arquivo com um cabeçalho descritivo e metadados, seguido pela seção de declarações e, então, o bloco principal.

### Modelo de cabeçalho (comentário)

/\*  
 Título: <Nome do Programa>  
 Autor: <Seu Nome>  
 Data: 2025-09-25  
 Versão: 1.0  
 Descrição: Breve descrição do objetivo do programa.  
 Entrada: O que o programa lê (ex.: dois inteiros).  
 Saída: O que o programa imprime (ex.: soma dos inteiros).  
 Observações: Regras, limitações ou referências.  
\*/

### Esqueleto de programa

algoritmo "NomeDoPrograma"  
 // Declarações (variáveis, constantes, funções, procedimentos)  
 var  
 inteiro a, b  
 real media  
 logico ativo  
 cadeia nome  
  
 const  
 real PI <- 3.14159  
  
 inicio  
 // Corpo principal  
 escreva("Digite um número: ")  
 leia(a)  
 escreva("Você digitou: ", a, "\n")  
 fim\_algoritmo

## Tipos, variáveis e constantes

* inteiro: números inteiros (…, -2, -1, 0, 1, 2, …)
* real: números reais (ponto flutuante)
* logico: verdadeiro ou falso
* cadeia: textos (strings)
* caractere: um único caractere

Declaração e atribuição:

var  
 inteiro idade <- 0  
 real altura <- 1.75  
 logico aprovado <- falso  
 cadeia nome <- "Ana"

Constantes:

const  
 real GRAVIDADE <- 9.80665

## Entrada e saída

* escreva(valores…): imprime valores em sequência
* escreval(valores…): imprime e quebra linha (se disponível na sua distribuição)
* leia(variavel): lê do teclado para a variável

escreva("Informe seu nome: ")  
leia(nome)  
escreval("Olá, ", nome)

## Operadores e expressões

* Aritméticos: +, -, \*, /, % (resto), ^ (potência, quando suportado)
* Relacionais: =, <>, >, >=, <, <=
* Lógicos: e, ou, nao

Precedência usual: potência > multiplicação/divisão > adição/subtração > relacionais > lógicos. Use parênteses para deixar claro.

inteiro a <- 5, b <- 2  
real r <- (a + 3) \* (b - 1) / 2  
logico ok <- (a > b) e nao (b = 0)

## Estruturas de seleção

### Se / Senão

se (idade >= 18) entao  
 escreval("Maior de idade")  
senao  
 escreval("Menor de idade")  
fim\_se

Encadeamento:

se (nota >= 9.0) entao  
 escreval("A")  
senao se (nota >= 7.0) entao  
 escreval("B")  
senao se (nota >= 5.0) entao  
 escreval("C")  
senao  
 escreval("D")  
fim\_se

### Escolha / Caso (switch)

inteiro opcao  
leia(opcao)  
  
escolha opcao  
 caso 1:  
 escreval("Cadastrar")  
 caso 2:  
 escreval("Listar")  
 caso 3:  
 escreval("Sair")  
 outrocaso:  
 escreval("Opção inválida")  
fim\_escolha

## Estruturas de repetição

### Para (contado)

inteiro i  
para i de 1 ate 10 passo 1 faca  
 escreval(i)  
fim\_para

Contagem regressiva:

para i de 10 ate 1 passo -1 faca  
 escreval(i)  
fim\_para

### Enquanto (pré‑condição)

inteiro x <- 0  
enquanto (x < 5) faca  
 escreval("x = ", x)  
 x <- x + 1  
fim\_enquanto

### Repita até (pós‑condição)

inteiro numero  
repita  
 escreva("Digite um número positivo: ")  
 leia(numero)  
ate (numero > 0)

## Funções e procedimentos

* procedimento: não retorna valor; usado para executar ações
* funcao: retorna um valor; pode receber parâmetros

procedimento cabecalho()  
 escreval("==== Sistema XYZ ====")  
 escreval("Autor: Você")  
fim\_procedimento  
  
funcao inteiro fatorial(inteiro n)  
 se (n <= 1) entao  
 retorne 1  
 fim\_se  
 retorne n \* fatorial(n - 1)  
fim\_funcao  
  
algoritmo "ExemploFuncoes"  
 var inteiro n  
 inicio  
 cabecalho()  
 escreva("n = ")  
 leia(n)  
 escreval("fatorial = ", fatorial(n))  
 fim\_algoritmo

Parâmetros por valor (padrão) e por referência (quando disponível) variam conforme distribuição. Se suportado, referencia antes do parâmetro indica passagem por referência.

## Vetores e matrizes

### Vetor (array unidimensional)

var  
 inteiro n <- 5  
 inteiro v[5]  
  
// leitura  
inteiro i  
para i de 0 ate n-1 passo 1 faca  
 escreva("v[", i, "] = ")  
 leia(v[i])  
fim\_para  
  
// impressão  
para i de 0 ate n-1 passo 1 faca  
 escreval("v[", i, "] = ", v[i])  
fim\_para

### Matriz (bidimensional)

var  
 inteiro lin <- 2, col <- 3  
 real m[2][3]  
  
inteiro i, j  
para i de 0 ate lin-1 passo 1 faca  
 para j de 0 ate col-1 passo 1 faca  
 m[i][j] <- i \* 10 + j  
 fim\_para  
fim\_para

## Strings e utilitários comuns

* Tamanho: tamanho(cadeia)
* Substring: subcadeia(txt, inicio, comprimento)
* Concatenação: operador & ou apenas , em escreva (varia por distribuição)
* Conversão: inteiro(cadeia), real(cadeia), cadeia(inteiro)

cadeia s <- "CompilAI"  
escreval("len = ", tamanho(s))  
escreval("sub = ", subcadeia(s, 0, 5))

## Boas práticas

* Nomeie variáveis de forma descritiva (totalVendas, indiceAluno).
* Prefira funções e procedimentos para modularizar.
* Evite números mágicos; use constantes.
* Comente o porquê, não o óbvio.
* Valide entradas do usuário.
* Indente blocos consistentemente.

## Exemplos completos

### 1) Calculadora simples (seleção)

/\* Calculadora de duas entradas \*/  
algoritmo "Calculadora"  
 var real a, b; cadeia op  
 inicio  
 escreva("a = "); leia(a)  
 escreva("b = "); leia(b)  
 escreva("Operação (+, -, \*, /): "); leia(op)  
  
 se (op = "+") entao  
 escreval(a + b)  
 senao se (op = "-") entao  
 escreval(a - b)  
 senao se (op = "\*") entao  
 escreval(a \* b)  
 senao se (op = "/") entao  
 se (b = 0) entao  
 escreval("Erro: divisão por zero")  
 senao  
 escreval(a / b)  
 fim\_se  
 senao  
 escreval("Operação inválida")  
 fim\_se  
 fim\_algoritmo

### 2) Fatorial (iterativo e recursivo)

funcao inteiro fatIter(inteiro n)  
 inteiro r <- 1, i  
 para i de 2 ate n passo 1 faca  
 r <- r \* i  
 fim\_para  
 retorne r  
fim\_funcao  
  
funcao inteiro fatRec(inteiro n)  
 se (n <= 1) entao  
 retorne 1  
 fim\_se  
 retorne n \* fatRec(n - 1)  
fim\_funcao

### 3) Fibonacci até N termos

procedimento fibonacci(inteiro n)  
 inteiro a <- 0, b <- 1, i, t  
 para i de 1 ate n passo 1 faca  
 escreva(a, " ")  
 t <- a + b  
 a <- b  
 b <- t  
 fim\_para  
 escreval("")  
fim\_procedimento

### 4) Verificador de número primo

funcao logico ehPrimo(inteiro n)  
 se (n < 2) entao retorne falso fim\_se  
 inteiro i  
 para i de 2 ate raiz(n) passo 1 faca  
 se (n % i = 0) entao retorne falso fim\_se  
 fim\_para  
 retorne verdadeiro  
fim\_funcao

### 5) Ordenação (Bubble Sort)

procedimento bubbleSort(inteiro v[], inteiro n)  
 inteiro i, j, tmp  
 para i de 0 ate n-2 passo 1 faca  
 para j de 0 ate n-2-i passo 1 faca  
 se (v[j] > v[j+1]) entao  
 tmp <- v[j]  
 v[j] <- v[j+1]  
 v[j+1] <- tmp  
 fim\_se  
 fim\_para  
 fim\_para  
fim\_procedimento

### 6) Busca binária (vetor ordenado)

funcao inteiro buscaBinaria(inteiro v[], inteiro n, inteiro alvo)  
 inteiro ini <- 0, fim <- n - 1  
 enquanto (ini <= fim) faca  
 inteiro meio <- (ini + fim) / 2  
 se (v[meio] = alvo) entao  
 retorne meio  
 senao se (v[meio] < alvo) entao  
 ini <- meio + 1  
 senao  
 fim <- meio - 1  
 fim\_se  
 fim\_enquanto  
 retorne -1  
fim\_funcao

## Exercícios propostos

1. Média de N números: leia N e os valores, calcule média e desvio padrão.
2. Jogo de adivinhação: gere número aleatório e peça tentativas com dicas.
3. Matriz transposta: leia uma matriz e imprima sua transposta.
4. Palíndromo: verifique se uma cadeia é palíndromo desconsiderando espaços.
5. Agenda simples: menu com incluir, listar e remover contatos (vetor de registros).

## Apêndice A: Tabela de palavras‑chave (referência rápida)

* Blocos: algoritmo, inicio, fim\_algoritmo
* Declarações: var, const, funcao, procedimento, retorne
* I/O: escreva, escreval, leia
* Seleção: se, senao, fim\_se, escolha, caso, outrocaso, fim\_escolha
* Repetição: para, fim\_para, enquanto, fim\_enquanto, repita, ate
* Tipos: inteiro, real, logico, cadeia, caractere
* Lógicos: verdadeiro, falso, e, ou, nao

## Apêndice B: Modelos prontos

### Modelo de cabeçalho

/\*  
 Título: <Título>  
 Autor: <Nome>  
 Data: <AAAA-MM-DD>  
 Versão: 1.0  
 Descrição: <O que o programa faz>  
 Entrada: <O que lê>  
 Saída: <O que imprime>  
\*/

### Modelo de função

funcao <tipo> nome(<parametros>)  
 // ...  
 retorne <valor>  
fim\_funcao

### Modelo de laços

// Para  
para i de <ini> ate <fim> passo <p> faca  
 // corpo  
fim\_para  
  
// Enquanto  
enquanto (<condicao>) faca  
 // corpo  
fim\_enquanto  
  
// Repita até  
repita  
 // corpo  
ate (<condicao>)

Dica: Se sua versão do CompilAI tiver diferenças de sintaxe (por exemplo, fim em vez de fim\_algoritmo, escreval ausente, ou índices começando em 1), adapte mantendo a lógica e a estrutura apresentadas aqui.