



# Desenvolvimento de Sistemas

---

**Algoritmos: naturais e estruturados; representações visuais, linguagem algorítmica, comandos de entrada, processamento e saída de dados**

Independentemente da linguagem de programação, do problema a ser resolvido ou do sistema a ser implementado, a lógica é a base para a construção de uma solução informatizada para uma necessidade, como você pôde conferir no conteúdo **Lógica de programação** desta unidade.

A lógica é a base inclusive para representações que facilitam a visão sobre o programa ou pode ser uma espécie de planejamento para programar em alguma tecnologia. Os algoritmos partem da lógica para esquematizar os passos necessários para se partir de uma situação original (entrada) e atingir um objetivo (saída) e podem ser representados de maneira mais livre e próxima da linguagem e do pensamento humano ou de maneira mais esquematizada, se aproximando de uma linguagem de programação, facilitando sua transcrição para a tecnologia escolhida.

É de grande importância que o programador planeje os passos a serem tomados antes de escrever um código, organizando suas ideias, apontando soluções e prevendo problemas, de maneira a repensar e adaptar os passos previstos.

Embora alguns programadores experientes dispensem essa etapa, especialmente para problemas menores e mais corriqueiros, a montagem de algoritmos é algo fundamental para uma boa programação, pois assim se pode pensar

na solução com mais propriedade, sem se preocupar com a tecnologia aplicada (abstraindo níveis de dificuldade e possíveis problemas). E ainda que experientes profissionais não escrevam em papel ou digitalmente um algoritmo mais formal, sem dúvida eles têm os passos já montados em sua mente.

Assim, para iniciantes em programação, a relevância de se escrever os algoritmos é maior, além de aplicar apenas o já treinado pelo pensamento, para que as soluções não sejam atreladas a uma tecnologia em específico, quebrando alguns “vícios” de programação.



## Algoritmos naturais e estruturados

Algoritmos são definidos como uma sequência finita de passos que levam à resolução de um problema. Forbellone e Eberspacher (2005) defendem que algoritmo é a sequência de passos que visam a atingir um objetivo bem definido. Xavier (2014) descreve algoritmo como uma sequência ordenada de passos que deve ser seguida para a realização de uma tarefa.

Conforme essas definições de algoritmos, pode-se criar as mais diversas rotinas para inúmeras situações. Por exemplo, a compra de um celular novo:

- 
- ◆ Analisar sua necessidade
  - ◆ Escolher a marca
  - ◆ Escolher o modelo
  - ◆ Escolher a cor
  - ◆ Negociar a forma de pagamento
  - ◆ Pagar
  - ◆ Retirar o celular

Perceba que não existe somente uma forma de realizar um algoritmo, você pode criar outros meios e sequências para obter o mesmo resultado.

Para compreender o funcionamento de um algoritmo, serão estudados algoritmos naturais e sequências.

## Algoritmos naturais

Os algoritmos naturais são criados de maneira simples e escritos de forma textual, usando o português coloquial. São claros, finitos e têm cada passo detalhado. Além disso, os algoritmos contribuem no desenvolvimento de uma aplicação, pois podem direcionar, de forma simples e eficiente, as descrições dos problemas e suas soluções.

Para reforçar os conceitos dos algoritmos naturais, veja dois exemplos.

Exemplo 1: como realizar a soma de dois números. Para isso, o usuário deverá entrar com os dois valores, e o computador retornará o resultado da soma com a operação aritmética soma.

1. Início
2. Entrar com o primeiro número

- 
3. Entrar com o segundo número
  4. Realizar a soma utilizando o primeiro número mais o segundo
  5. Mostrar na tela o resultado da soma
  6. Fim

Exemplo 2: etapas de como trocar um pneu furado.

1. Início
2. Afrouxar ligeiramente as porcas
3. Suspender o carro
4. Retirar as porcas e o pneu
5. Colocar o pneu reserva
6. Apertar as porcas
7. Abaixar o carro
8. Dar o aperto final nas porcas
9. Fim

Os algoritmos naturais são poucos usados na prática, pois, muitas vezes, a linguagem coloquial dá oportunidade a más interpretações e imprecisões, como, por exemplo, a instrução “afrouxar ligeiramente as porcas”, que, no algoritmo, está sujeita a diferentes interpretações de pessoas. Uma instrução mais precisa seria “afrouxar as porcas, girando-as 30° no sentido anti-horário”.

Além disso, para transcrever os algoritmos naturais em linguagem computacional, necessita-se de um prévio entendimento, que posteriormente poderá sofrer transformações em forma de algoritmos, diagramas em blocos (fluxogramas ou diagrama de Chapin) e pseudocódigos.

## Algoritmos estruturados

Os algoritmos estruturados se aproximam de uma linguagem de programação convencional. Esses algoritmos são criados seguindo uma estrutura de comandos, sintaxes e termos específicos. Além disso, apresentam estruturas bem mais definidas, como:

- ◆ **Sequências:** instruções executadas uma a uma, na sequência em que aparecem no programa.
- ◆ **Seleções:** instrução *if-else*, que seleciona 1 entre 2 caminhos possíveis.
- ◆ **Repetições:** instruções executadas várias vezes.

Conforme visto na definição, algoritmo é uma sequência finita de instruções que, ao serem executadas, chegam à solução de um problema. Sendo assim, a sequência de instruções necessita que, em algum momento, o algoritmo seja finalizado.

Veja a seguir o esqueleto de um algoritmo estruturado que apresenta uma sequência de comandos (instruções).

Início

- ◆ <comando1>
- ◆ <comando2>
- ◆ <comando “n”>

Fim

## Representação visual de algoritmos

Algoritmo é uma linha de raciocínio que pode ser escrita de várias formas, seja textual, seja graficamente. Até este momento, os algoritmos foram representados de forma textual, usando o português coloquial. Entretanto, é possível representar

algoritmos com figuras para ações específicas que serão executadas pelo computador. Essas representações gráficas podem ser utilizadas para representar algoritmos pequenos ou de tamanho médio para facilitar o entendimento.

Veja a seguir dois dos métodos mais empregados para representação visual de algoritmos: diagrama de blocos (fluxograma) e diagrama de Nassi-Shneiderman (Chapin).

## Diagrama de blocos (fluxograma)

O diagrama de blocos ou fluxograma é uma técnica de representação gráfica que utiliza símbolos, permitindo a descrição de uma sequência, ou um fluxo, e de um processo determinado pelo programador para a resolução de problemas.

O fluxograma determina a linha de raciocínio utilizada pelo programador para resolver problemas. Ao escrever um fluxograma, o programador deve estar ciente de que os símbolos utilizados devem estar em harmonia e ser de fácil entendimento. Para que os diagramas de blocos tenham certa coerência, os seus símbolos foram padronizados pelo Instituto Norte-Americano de Padronização (ANSI).

Veja a definição dos principais símbolos utilizados no fluxograma, baseados na notação do software Flowgorithm, que será utilizado durante os demais conteúdos do curso.



	Representa o início do fluxograma
	Representa o fim do fluxograma
	Declaração de variáveis
	Operação com entrada de dados
	Operação com saída de dados
	Operações de atribuição
	Algoritmos que necessitam de estrutura de decisão

Tabela 1 – Símbolos de fluxograma no Flowgorithm

o primeiro item é uma elipse com texto com o rótulo “Principal”, representando o início do fluxograma. O segundo item é uma elipse com o rótulo “Fim”, representando o fim do fluxograma. Na sequência, há uma caixa retangular com o rótulo “Declarar”, com a declaração das variáveis. Em seguida, há um paralelogramo com “Ler” para as operações de entrada de dados. Após, outro paralelogramo com “Escrever” para as

operações de saída de dados, seguido de uma caixa retangular com o rótulo “Atribuir” para as operações de atribuição. Por fim, um losango com “Se” de onde partem duas setas: uma escrita “Falso” e outra “Verdade” para algoritmos que necessitam de uma estrutura de decisão.

Como exemplo, será realizado um fluxograma de algoritmo que solicita dois números, realiza a soma e exibe na tela. Para um melhor entendimento, acompanhe os comentários de execução para cada símbolo.

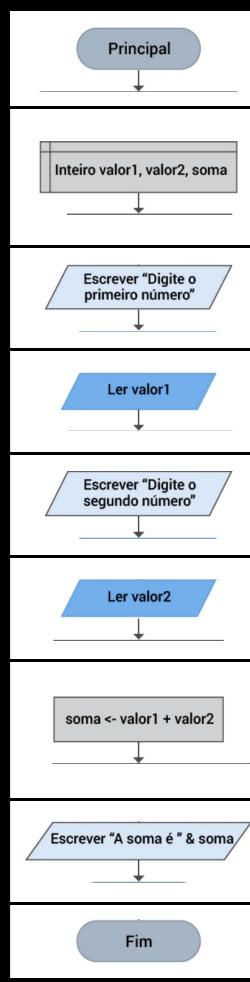
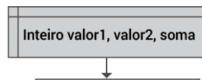
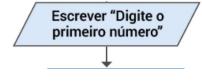
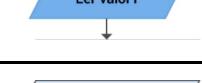
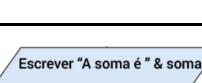
	O símbolo deu início ao diagrama de blocos.
	O símbolo de declaração de variáveis definiu que serão utilizados no diagrama valor1, valor2, soma.
	O símbolo de saída mostra na tela o que o usuário deve fazer.
	O símbolo de entrada de dados libera para o usuário entrar com o primeiro número.
	O símbolo de saída mostra na tela o que o usuário deve fazer.
	O símbolo de entrada de dados libera para o usuário entrar com o segundo número.
	O símbolo é realizado com as atribuições e os valores calculados para suas respectivas variáveis.
	O símbolo de saída mostra na sua tela o resultado do valor calculado.
	Finaliza o programa.

Tabela 2 – Algoritmo em fluxograma para a soma de dois números

O fluxograma inicia com o símbolo “Principal”, que leva ao símbolo “Declaração”, com os termos “íntero valor1, valor2 e soma”. Após, o símbolo de “Escrever”, com o comando “Digite o primeiro número”, levando ao símbolo “Ler”, que armazena em “valor1” o valor digitado pelo usuário. Novamente aparece o símbolo “Escrever”, agora com o comando “Digite o segundo número”, levando ao símbolo “Ler”, que armazena em “valor2” o segundo valor digitado pelo usuário. Em seguida, o símbolo de atribuição no qual será atribuído no termo “soma”, o conteúdo armazenado em “valor1” mais “valor2”. Na sequência, há o símbolo “Escrever”, que exibe a soma dos dois números armazenados no termo “soma”. Por fim, a elipse com rótulo “Fim” finaliza a execução do fluxograma.

Anteriormente foi apresentado o termo “variável”. Variáveis são espaços na memória onde guardamos uma informação para usá-la posteriormente. Veja mais sobre isso neste conteúdo e no texto “Manipulação de dados” desta unidade curricular.

## Diagrama de Nassi-Shneiderman (Chapin)

O diagrama de Nassi-Shneiderman ou Chapin permite a visualização do fluxo lógico do algoritmo e é voltado para a programação estruturada. É uma ferramenta de representação gráfica que oferece grande clareza na representação de algoritmos sequenciais, de seleção e repetição.

O diagrama utiliza uma simbologia própria, em que a ideia básica é representar as ações de um algoritmo dentro de um único retângulo, subdividindo-o em retângulos menores que representam os diferentes blocos de sequência de ações do algoritmo, como:

- ◆ **Início:** indica que o algoritmo iniciou
- ◆ **Escreva:** exibe uma informação na tela
- ◆ **Leia:** aguarda a inserção de uma informação
- ◆ **Fim:** indica o final da execução do algoritmo

Veja a simbologia utilizada no diagrama de Chapin:



## Bloco das sequências

Quando uma ação é executada e avança para o próximo bloco.

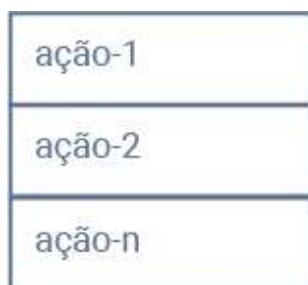


Figura 1 – Bloco de sequência

Um retângulo dividido em três linhas. Em cada linha está escrito ação-1, ação-2 e ação-n.

## Bloco de seleção

Existem dois tipos de blocos de ramificação. A primeira ramificação é verdadeira ou falsa, dependendo do cumprimento ou não da condição.

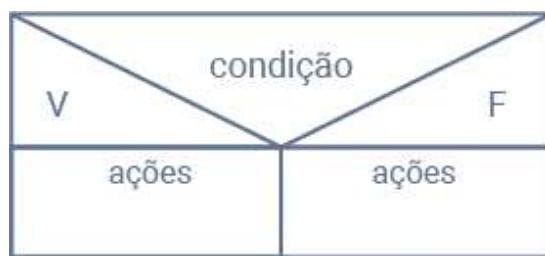


Figura 2 – Bloco de seleção

Um retângulo escrito “condição” na parte central superior dividido em duas direções: “verdadeira” com ações próprias e “falsa” também com ações próprias.

Observe três exemplos do diagrama de Chapin a seguir:

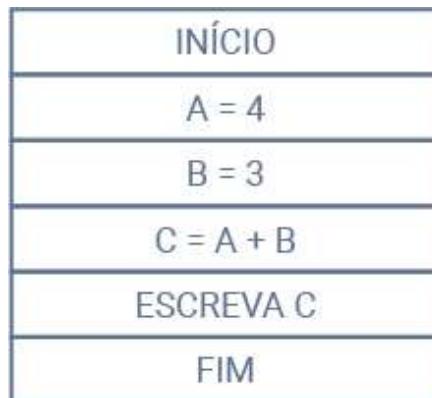


Figura 3 – Algoritmo de soma de dois números fixos em Chapin

Um retângulo dividido em seis linhas. Na primeira, a palavra “início”, na segunda, a atribuição do valor 4 para a variável “A”. Na terceira linha, a atribuição do valor 3 para variável “B”. A quarta linha contém a variável “C” com a soma da variável “A” com a “B”. Na penúltima linha, o comando “Escreva” exibe o valor da variável “C”. Na sexta e última linha, há o termo “Fim”.

Nesse primeiro exemplo, há um algoritmo bastante simples, que tem a função de somar dois números e informar seu resultado na tela. No algoritmo, é definido o valor de “4” para a variável “A” e o valor “3” para a variável “B”. Todo o cálculo é feito na variável “C” pelo operador aritmético soma (+). Na sequência, é exibido o conteúdo presente na variável “C” pelo comando **Escreva** seguido da finalização.

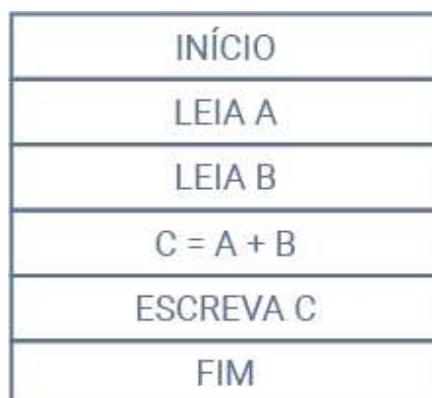


Figura 4 – Algoritmo que solicita a entrada de dois valores para soma

Um retângulo dividido em seis linhas: na primeira, há o comando “Início”, na segunda, o comando que solicita que seja informado um valor para ser atribuído na variável “A”. Na terceira linha, há o comando que solicita que um valor para ser atribuído na variável “B” seja informado. Na quarta linha, na variável “C”, é atribuída a soma da variável “A” com a “B”. Na penúltima linha, o comando “Escreva” exibe o valor da variável “C”. Na sexta e última linha, há o termo “Fim”.

No segundo exemplo, o algoritmo anterior foi aprimorado com comandos para que o usuário informe quais valores ele quer definir para A e para B no algoritmo. Veja que todas as ações descritas são sequenciais. O diagrama inicia, lê os dois números, armazena-os nas variáveis A e B, calcula o valor e armazena-o na variável C, escreve o valor de C e finaliza o diagrama. Esse é um algoritmo de soma dinâmico, que solicita a entrada de valores e a exibição do cálculo.

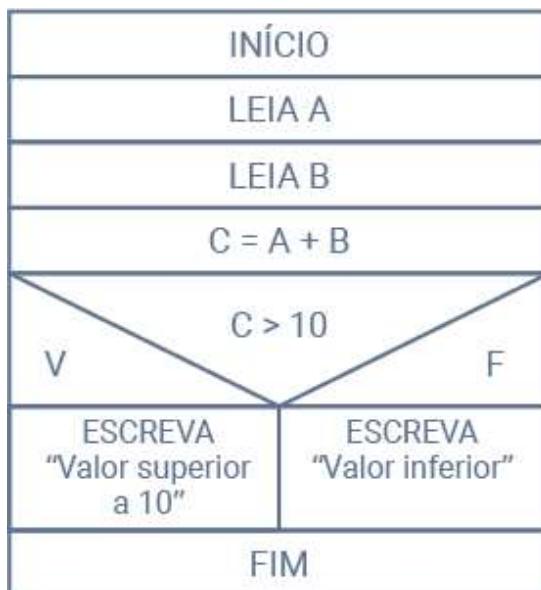


Figura 5 – Algoritmo em Chapin com bloco de seleção

Um retângulo dividido em sete linhas: na primeira, há o comando “Início”, na segunda, o comando que solicita que seja informado um valor para ser atribuído na variável “A”. Na terceira linha, o comando solicita que seja informado um valor para ser atribuído na variável “B”. Na quarta linha, na variável “C”, é atribuída a soma da

variável “A” com a “B”. Na quinta linha, há uma condição com duas direções: se o valor contido na variável “C” for superior a 10, então será exibida a mensagem “Valor superior a 10”, caso contrário, será exibida a mensagem “Valor inferior”. Na sexta e última linha, há o termo “Fim”.

No terceiro exemplo, foi adicionado um bloco de seleção ao algoritmo anterior, informando o usuário caso o valor da soma seja superior a “10” (lado “V” ou “Verdadeiro”), então será exibido na tela a mensagem “Valor superior a 10”, caso contrário (lado “F”, ou “Falso”), “Valor inferior”.

## Linguagem algorítmica

Tanto os algoritmos naturais quanto as representações visuais de algoritmos permitem um nível de clareza muito grande, porém eles têm suas vantagens e desvantagens. A facilidade de perceber o fluxo de execução do programa é evidentemente mais clara, mas sua grande desvantagem é que existe a necessidade do conhecimento das convenções gráficas de cada uma das técnicas, pois, apesar de serem simples, não são naturais, pois é mais comum a expressão de ações por meio de palavras.

Por isso o uso de algoritmos textuais, que são mais naturais e, por consequência, mais simples. Para representar algoritmos textualmente, será usado o português, como explicado anteriormente.

## Pseudocódigo

O pseudocódigo é considerado uma ferramenta que pode auxiliar a programação. Pode ser escrito em palavras similares ao inglês ou ao português para facilitar a interpretação e o desenvolvimento de um programa. Pseudocódigo **também é chamado de linguagem estruturada**, pois reúne um conjunto de comando ou ações,

usando uma linguagem simples, que se aproxima de uma linguagem de programação. Além disso, obedece a regras de sintaxe, por exemplo, como cada termo é escrito, quais palavras formam determinados comandos, como e onde usar pontuações, espaços e palavras reservadas etc.

Existem várias formas de representar um pseudocódigo, mas o mais importante é que ele apresente a solução do problema proposto.

Veja agora o pseudocódigo em Portugol, para resolver o algoritmo da soma de dois números citados anteriormente.

```
programa
{
    Função início()
    {
        inteiro valor1,valor2,soma

        valor1 = 4
        valor2 = 3

        soma = valor1 + valor2

        escreva("A soma dos números é igual a: ", soma)
    }
}
```

## Portugol

É uma pseudolinguagem algorítmica ou método de codificação que tem por objetivo ensinar estruturas básicas para escrever algoritmos, utilizando a língua portuguesa.

Portugol é um acrônimo formado pelas palavras **Português + algol + Pascal**, também conhecido como português estruturado, criado por Antônio Carlos Nicolodi e Antônio Manso, em 1986.

Existem diversas ferramentas para aprendizado de algoritmos em Portugol, como, por exemplo: VisualG, Portugol IDE e Portugol Studio. Por não se tratar de uma linguagem de programação propriamente dita, podem existir variações na codificação do Portugol nessas ferramentas.

Neste curso, será adotado o **Portugol Studio**, por ser uma das mais completas ferramentas da área e ser possível utilizá-la tanto no computador como também *online*, diretamente no navegador de internet. É importante destacar também que o Portugol Studio é apenas uma ferramenta e que, no dia a dia, o algoritmo pode ser feito até à mão, com papel e caneta.

## Portugol Studio



 O Portugol Studio é um Ambiente de Desenvolvimento Integrado (IDE – *Integrated Development Environment*) completo para Portugol e um dos mais simples de usar. Sua primeira versão começou a ser utilizada em 2011 na Universidade do Vale do Itajaí (UNIVALI), instituição do Laboratório de Inovação Tecnológica na Educação (LITE), onde a plataforma foi criada por Luiz Fernando Noschang, graduando de ciência da computação na época.

Dois anos depois, a ferramenta chegou à sua versão 2.0 e começou a ser utilizada em outras universidades, como a Universidade Federal do ABC, e um ano depois alcançou a versão 2.2. Desde seu início até o momento, o Portugol Studio conta com membros ativos que mantêm a ferramenta.

A instalação é bastante simples, basta entrar no site do projeto, clicar no botão **Download** para baixar a ferramenta e continuar com a instalação. O Portugol Studio é multiplataforma: tem versões para Windows, Mac e Linux.

- ◆ Endereço para baixar: <http://lite.acad.univali.br/portugol/>  
(<http://lite.acad.univali.br/portugol/>)

## PortugolWebstudio

O PortugolWebstudio nasceu como uma adaptação para o Portugol Studio e a necessidade de torná-lo acessível em qualquer lugar, ou seja, disponível para navegadores, sem a necessidade de instalação no seu sistema operacional.

- ◆ Endereço de acesso: <https://portugol-webstudio.cubos.io/>  
(<https://portugol-webstudio.cubos.io/>)

## Conhecendo o Portugol

Agora que você já instalou ou acessou de forma *on-line* pelo PortugolWebstudio, você terá na parte superior da tela, abaixo das abas, ícones com funcionalidades para criar e abrir arquivos, ajuda, entre outros. No lado direito da tela, há uma árvore com diversos exemplos. Clicando no exemplo, você verá uma descrição e um botão para explorá-lo.



Figura 6 – Portugol Studio

Uma tela do Portugol Studio na parte superior. Abaixo das abas, ícones de funcionalidades para criar um arquivo com o botão de “mais”. O próximo ícone é para abrir um arquivo com botão de pasta aberta. Na sequência, um ícone com interrogação para ajuda e, por fim, o ícone “Plugins”. Já no lado direito da tela há uma árvore com diversos exemplos com um botão de busca para pesquisar nos exemplos disponíveis.

No ícone **+** **Novo Arquivo**, você criará seus algoritmos em Portugol. Já o ícone de **? Ajuda** pode ser utilizado para consultar alguma dúvida sobre a sintaxe usada no Portugol.

## Criando “Olá Mundo” no Portugol Studio

Depois da visão geral, chegou o momento de você criar seu primeiro programa no Portugol Studio. Primeiramente, abra o Portugol e crie um arquivo. Como você verá, ele já tem uma estrutura inicial, que é o “Olá Mundo” em português.

No lado esquerdo da tela, você terá alguns ícones, como **Play** (ou aperte **Shift + F6** no teclado), para executar o algoritmo. O texto “Olá Mundo”, que o comando **Escreva()** pede para escrever, é descrito no *console* (veja na parte inferior da imagem a seguir). Por fim, salve seu algoritmo clicando no ícone **Pen drive** (ou apertando **Ctrl + S** no teclado).

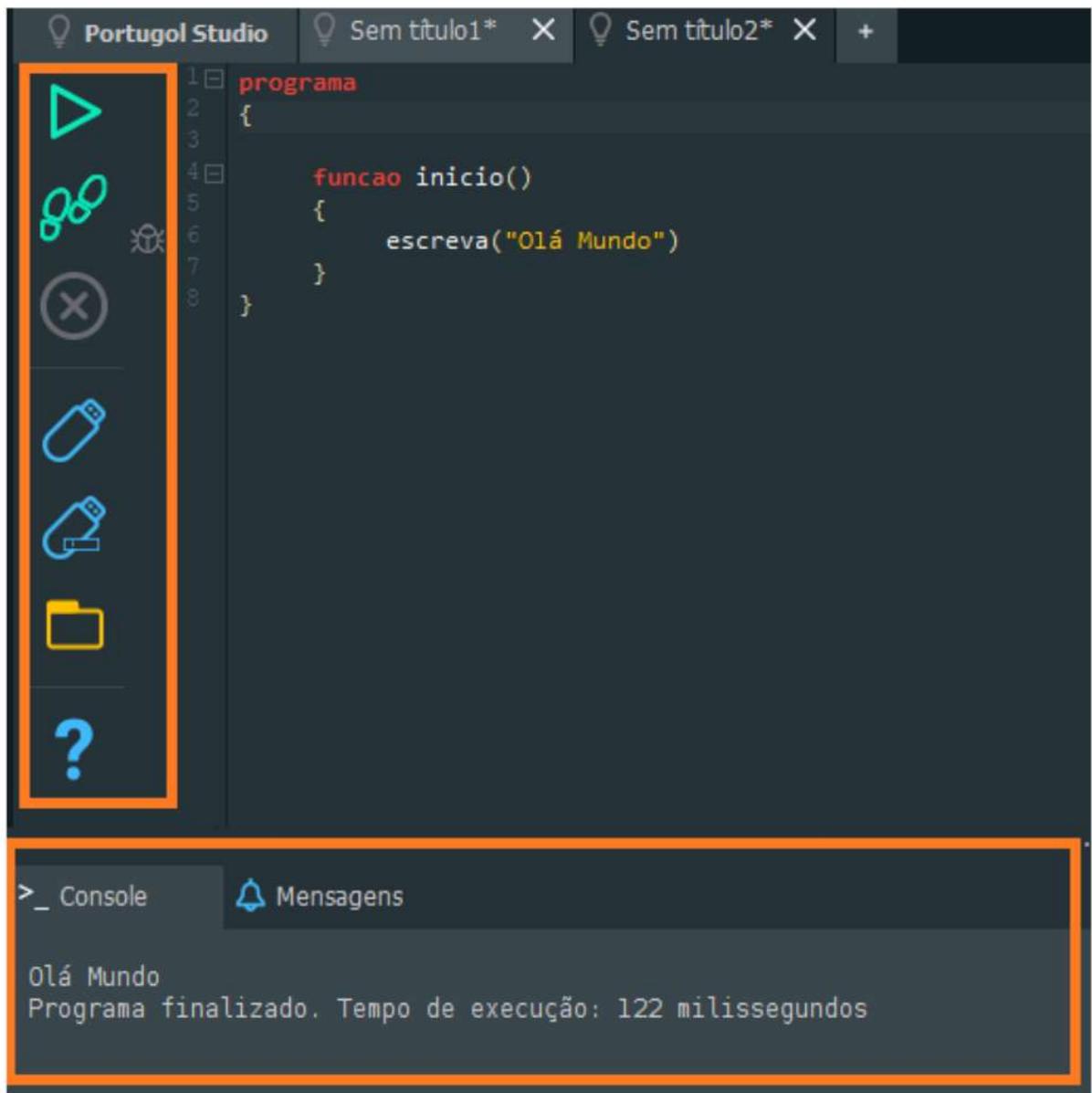


Figura 7 – Portugol Studio: criando seu primeiro “Olá Mundo”

Na tela do Portugol Studio, há no centro a estrutura básica de um algoritmo em Portugol para escrever na tela “Olá Mundo”. No menu esquerdo, há os comandos para a execução do algoritmo. O botão de “Play” do lado esquerdo deverá ser acionado ou

as teclas de atalho do teclado “Shift + F6” para a execução do algoritmo. Para salvar o algoritmo, há um botão de “Pen drive” do lado esquerdo. Por fim, há um ícone de “interrogação” para dúvidas. Na parte inferior da tela, há duas abas: “Console”, em que serão exibidas mensagens e também digitados valores quando solicitado no algoritmo, e a aba “Mensagens”, em que serão exibidos alertas quando necessário.

## Estrutura de um algoritmo em Portugol Studio

Todos os algoritmos no Portugol precisam estar dentro da função **início()** que, por sua vez, precisa estar dentro do bloco **programa**. Isso será comum a todos os exemplos seguintes, então será adotado apenas o pseudocódigo que fica dentro da função **início()** nos exemplos deste conteúdo.

```
programa
{
    Função início()
    {
        //bloco das declarações (variáveis)

        //bloco dos comandos (entrada, processamento e saída)
    }
}
```

Conforme esse bloco de código, os algoritmos são organizados no bloco das declarações e comandos. No início de algumas linhas, há “//”. Essas linhas têm por objetivo apenas dar orientações e criar observações para quem está escrevendo ou lendo o algoritmo. Para inserir um comentário, é necessário inserir “//” no início da linha, e tudo que for escrito nela, até o seu final, será ignorado. Caso exista a necessidade de escrever algum comentário mais longo, devem ser utilizadas diversas linhas, lembrando-se de digitar “//” no início de cada uma.

## Bloco de declarações

É definida a lista de identificadores que serão usados na execução do algoritmo. Os identificadores ou símbolos são nomes que podem ser fornecidos para variáveis em um programa.

Exemplo 1: o identificador “valor1” associado ao tipo de dado “inteiro”

```
//bloco das declarações (variáveis)  
inteiro valor1
```

Exemplo 2: quando é necessário associar mais de um identificador ao mesmo tipo de dados

```
//bloco das declarações (variáveis)  
inteiro valor1, valor2, soma
```

Regras para a criação de identificadores:

- ◆ Podem ser usados letras, números e o caractere \_ (*underscore*).
- ◆ Sempre começar por letras ou pelo caractere \_ (*underscore*).
- ◆ Ao usar letras, podem ser maiúsculas ou minúsculas. O Portugol é *case sensitive*, ou seja, letras maiúsculas e minúsculas são diferentes.

Exemplo: “senac” é diferente de “Senac”.

- ◆ Não podem ser usados símbolos, como “\$, #, ?, !, &, >, <, +,-”.
- ◆ Nunca usar espaços em branco e caractere com - (hífen).
- ◆ Não usar palavras reservadas.

As palavras reservadas são identificadores predefinidos com significados especiais para o interpretador do algoritmo. Essas palavras variam de acordo com a linguagem utilizada. No Portugol, temos as seguintes palavras:

Programa	função	Início
Inteiro	real	Cadeia
Escreva	leia	Se
Senão	const	Inclua

## Bloco de comandos

Neste bloco, há os comandos de entrada e saída de dados e operadores de atribuição e aritméticos, que são os meios pelos quais se realizam as operações sobre as variáveis.

- ◆ **Operadores de atribuição:** usados para atribuir valores para os identificadores ou operações entre eles.

Exemplo: atribuir o valor 4 para o identificador “valor1” e 3 para o identificador “valor2”.

```
//bloco dos comandos
valor1 = 4
valor2 = 3
```

- ◆ **Operadores aritméticos:** adição (+), multiplicação (\*), subtração (-) e divisão (/).

Exemplo: é atribuído no identificador “soma” a adição do identificador “valor1” e “valor2”.

```
soma = valor1 + valor2
```

- ◆ **Saída de dados:** definida pelo comando **escreva()**, é usada para imprimir algo na tela, como visto no exemplo para imprimir “Olá mundo”.

É possível aprofundar-se um pouco mais no comando usando `\n` para pular uma linha. Como exemplo, será adaptado o “Olá Mundo”.

```
escreva("Olá mundo\nSENAC")
```

Será impresso:

Olá mundo  
SENAC

Você também pode escrever variáveis no comando **escreva()**:

```
valor1 = 4
valor2 = 3

soma = valor1 + valor2

escreva("A soma dos números é igual a ", soma, "!")
```

Será impresso:

A soma dos números é igual a 7!

Nesse código, as variáveis “valor1” e “valor2” foram inicializadas com os valores 4 e 3, respectivamente. Depois, foi usado o comando **escreva()** para imprimir a soma desses dois valores. Após colocar a variável “soma” dentro do **escreva()**, foi possível concatenar outro texto, nesse caso, o sinal de exclamação. Esse recurso é bastante usado nos algoritmos.

- ◆ **Entrada de dados:** definida pelo comando **leia()**, corresponde a uma entrada de dados por meio do dispositivo de entrada do computador (teclado). Esse comando só aceita variáveis como argumento. O que o usuário digitar será armazenado nas variáveis.

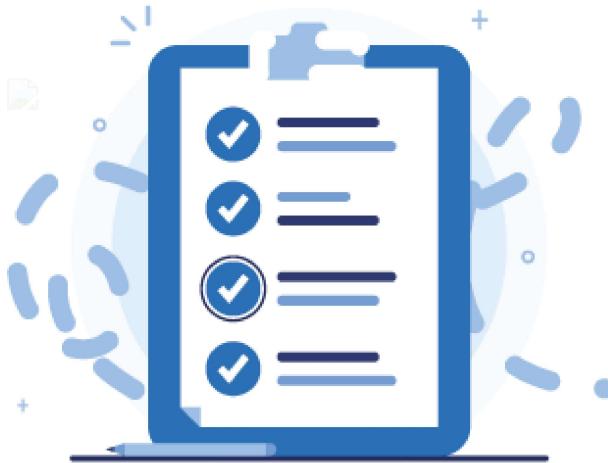
No exemplo a seguir, é solicitado, pelo comando **escreva()**, que o usuário informe seu primeiro número. O cursor ficará aguardando. Quando o usuário digitar o conteúdo, ficará armazenado na variável “valor1”.

```
inteiro valor1

escreva("Informe o primeiro número ")
leia(valor1)

escreva("Você informou ",valor1," como primeiro número")
```

Quando for referido o uso de pseudocódigos em algoritmos do tipo Portugol, significa que esse deve ser escrito de forma que outras pessoas possam entender de maneira mais formal, ou seja, os conceitos devem ficar claros e em uma linguagem acessível para quem interpretar o algoritmo.



Regras básicas importantes na utilização de pseudocódigos:

- ◆ Escolher um nome.
- ◆ Avaliar as variáveis e dar atenção a seus tipos e suas características.
- ◆ Descrever, de forma clara, o que será armazenado e verificar se as variáveis destinadas a essa informação estão corretas.
- ◆ Verificar se as instruções fazem sentido e se têm uma sequência lógica.
- ◆ Avaliar o resultado e, quando pertinente, mostrar na tela.
- ◆ Finalizar o algoritmo

## Comandos de entrada, processamento e saída

Para descrever o funcionamento de um algoritmo, você pode usar como base a representação do funcionamento do computador na arquitetura de John Von Neumann, baseado no modelo de entrada, processamento e saída.

**ENTRADA → PROCESSAMENTO → SAÍDA**

Figura 8 – Fluxo de um algoritmo com entrada, processamento e saída

Fluxo de um algoritmo descrevendo “Entrada”, seta para direita, “Processamento”, seta para direita e “Saída”.

Nesse modelo, as entradas dos algoritmos são caracterizadas pelos dados a serem fornecidos inicialmente. Já o processamento será definido pela execução das ações sobre os dados de entrada e sobre os dados intermediários gerados durante a execução do processamento. Por fim, a saída será a solução do problema ou o objetivo esperado.

Com base nesse modelo, veja um exemplo completo em Portugol de um algoritmo para calcular a soma de dois números. O algoritmo deverá solicitar o primeiro e o segundo número, realizar o processamento e exibir como saída o resultado da soma.

```
1 programa
2 {
3
4     função início()
5     {
6         inteiro valor1, valor2, soma
7
8         escreva("Digite o primeiro número: ")
9         leia(valor1) //ENTRADA: informa valor1
10
11        escreva("Digite o segundo número: ")
12        leia(valor2) //ENTRADA: informa valor2
13
14        soma = valor1 + valor2 // PROCESSAMENTO: Soma os dois valores
15
16        escreva("A soma dos números é igual a: ", soma) // SAIDA: Exibe o resultado
da soma
17    }
18 }
```

Veja os comentários desse algoritmo:

- ◆ Linha 6: inicia com o tipo de informação que as variáveis operacionalizarão no algoritmo, nesse caso, de números inteiros, seguidos dos nomes das variáveis “valor1”, “valor2” e “soma”.
- ◆ Linha 8: **escreva()** é um comando de saída e indica o que será mostrado na tela do computador. Geralmente o conteúdo do texto fica entre aspas duplas (“Digite o primeiro número”).
- ◆ Linha 9: **leia()** é um comando de entrada, e o valor digitado será armazenado na variável “valor1”.
- ◆ Linha 11: **escreva()** é um comando de saída e indica o que será mostrado na tela do computador. Geralmente o conteúdo do texto fica entre aspas duplas (“Digite o segundo número”).
- ◆ Linha 12: **leia()** é um comando de entrada, o valor digitado será armazenado na variável “valor2”.
- ◆ Linha 14: é o cálculo da soma dos números armazenados nas variáveis “valor1” e “valor2”. Esse cálculo é atribuído para a variável “soma”.
- ◆ Linha 16: o comando **escreva()** exibe na tela o que está entre aspas duplas (“\n A soma dos números é igual a: ”, soma) e o valor que está armazenado na variável soma. A variável é colocada fora das aspas, para que ela seja representada pelo seu valor correspondente.

Dessa forma, comprehende-se que a entrada de dados ocorre nas linhas 9 e 12 do algoritmo, quando o usuário informa os valores dos dois números. Já o processamento ocorre na linha 14, quando ocorre a soma dos valores e a atribuição para a variável soma. Por fim, a saída, ou seja, o resultado esperado pelo algoritmo quando ele mostra o valor processado.

## Encerramento

Foi possível concluir, neste conteúdo, que o uso de pseudocódigo é muito importante no processo de desenvolvimento de algoritmos. O Portugol é um dos mais conhecidos pseudocódigos, é bastante rico em detalhes e muito útil por assemelhar-se bastante à forma na qual os programas são escritos. A utilização de Portugol permite

ao programador expressar as suas ideias sem ter que se preocupar com a sintaxe da linguagem de programação. Além disso, foram observados em detalhes os blocos de declarações e os comandos da estrutura em Portugol, necessários para a concepção de algoritmos com entrada, processamento e saída.

Assista ao vídeo a seguir sobre linguagem algorítmica e comandos de entrada, processamento e saída.



