

Lógica de Programação: começando a desenvolver seus primeiros programas Turma 2024A

Descrição do Curso

Neste curso iremos desbravar nossas primeiras experiências no desenvolvimento de programas. Para isso, vamos utilizar o software Portugol Studio, que permite construir programas de computadores utilizando comandos em língua portuguesa (molezinha, né?). A partir dele, vamos compreender o conceito de entrada e saída de dados, variáveis e operadores!

Gustavo Pires Bertaco
Julho 2024

Sumário

1	Variáveis, Saída e Entrada de Dados e Operadores Aritméticos	2
1.1	Introdução	2
1.2	O que é programa de computador? O que é o Portugol Studio? .	2
1.3	Link para Download do Portugol Studio	4
1.4	Meu primeiro programa	4
1.5	Vídeo de introdução ao Portugol Studio	5
1.6	Variáveis	5
1.7	Saída de Dados	7
1.8	Entrada de Dados	8
1.8.1	Exemplo: Número Digitado	9
1.8.2	Exemplo: Nome	10
1.9	Operadores Aritméticos	10
1.9.1	Exemplo: Quatro Operadores	11
2	Vídeo sobre Entrada e Saída de Dados, Variáveis e Operadores Aritméticos no Portugol Studio	12
2.1	Praticando um pouco...	13
2.2	Solucionando Erros	14
2.2.1	Mensagens de Erro	15
2.2.2	Testes de Mesa	19

1 Variáveis, Saída e Entrada de Dados e Operadores Aritméticos

1.1 Introdução

Todo programa de computador deve obedecer uma estrutura definida por sua linguagem de computador. Mas... por quê? Basta pensarmos assim: como você faz para se comunicar com uma pessoa? Se você for uma pessoa bem educada, você primeiro irá cumprimentá-la:

- Bom dia! (ou boa tarde, boa noite...)

Depois disso, vocês baterão um papo legal. Ao final, provavelmente vocês se despedirão:

- Tchau!

Claro que há muitas formas de você ter esse bate papo, bem como se cumprimentar. Porém, de forma geral, as pessoas se cumprimentam, conversam e se despedem.

Mas agora, te pergunto: e se você tivesse que conversar com alguém em inglês? Ih... tá, talvez você não seja o expert (ou nem goste) em falar inglês, mas vamos para uma aulinha básica. Provavelmente, você falará:

- Hi! (tradução: Olá!)

(papo vai, papo vem...)

- Bye! (tradução: Adeus!)

Veja que a forma ou estrutura de comunicação é igual, o que muda é o idioma. Ora falamos em português, ora em inglês. E, o que isso tem a ver com os computadores? Bom, se você não captou ainda, os computadores também têm a sua linguagem! Aliás, os programas têm a sua linguagem! Ou seja, há as linguagens de programação. Cada linguagem, assim como os idiomas, tem semelhanças e particularidades (como a palavra "saudades", que só existe na língua portuguesa).

Bom, se quisermos programar um software, podemos usar as linguagens C++, Java, Pascal... Se quisermos programar para web, podemos usar PHP, ASP, JSP. E assim por diante. E aqui, não será diferente: vamos utilizar uma linguagem para programar nossos primeiros códigos. Ela será o Portugol!

Essa linguagem é fantástica! Pois utilizamos comandos baseados na língua portuguesa para nos comunicarmos com o computador. E não se preocupe, se você precisar programar em outras linguagens, bastará traduzir os principais comandos para a linguagem de programação desejada. Afinal, assim como no português, a estrutura será bem semelhante, somente temos que aprender novas palavras.

1.2 O que é programa de computador? O que é o Portugol Studio?

Antes de começar os estudos, vamos conhecer o Portugol Studio.

O Portugol Studio é um software que auxilia na aprendizagem da programação de computadores. Ele se assemelha bastante com os softwares reais de

programação, tendo como sua principal vantagem a simplicidade e o uso de comandos na língua portuguesa (o que facilita em muito o entendimento dos primeiros códigos).

Mas o que é código-fonte, programa, algoritmo?? Bom, é tudo quase a mesma coisa... um programa de computador é um conjunto de comandos que será realizado pelo computador. Esse conjunto de comandos deve seguir um raciocínio lógico, o que na matemática é chamado de algoritmo. Esse algoritmo é algo mais alto nível, algo que você pode explicar para seus avós ou para seu irmão mais novo, que eles vão entender. Agora, para que seu computador entenda, é preciso que o algoritmo seja escrito em forma de código-fonte. Ou seja, no fundo, no fundo, tudo é a mesma coisa!

Agora, vamos começar! Primeiro, faça o download e instale o Portugol Studio, pois nada como aprender algo na prática, certo?

• 1-Fazendo o download e instalando o Portugol Studio

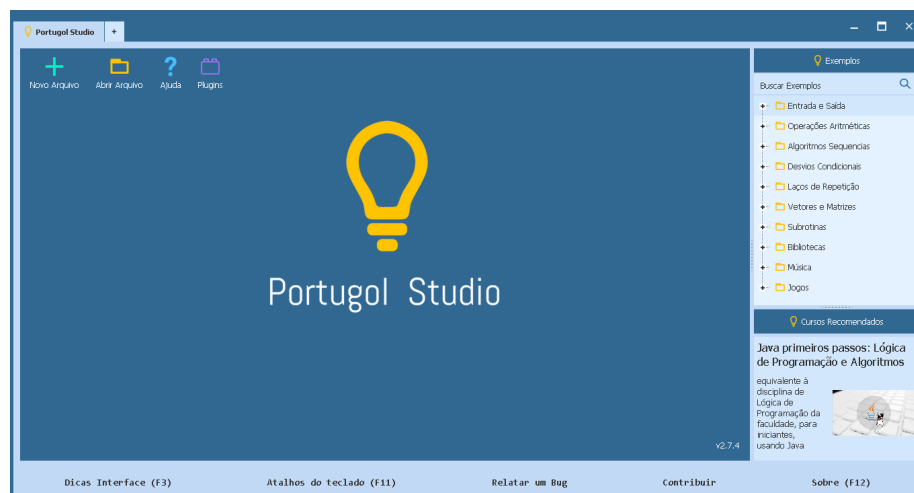
Você pode encontrar o link para baixar o Portugol Studio na página principal do nosso curso, ou acessando em o link Download em: <http://lite.acad.univali.br/portugol/>

Após baixar, basta fazer a instalação do software.

Atenção: o Portugol Studio é compatível com Windows, Linux e MAC OS. O tamanho do arquivo para download é de aproximadamente 75 MB. Dúvidas ou problemas na instalação, acesse <http://lite.acad.univali.br/portugol/> ou entre em contato com portugol.studio@gmail.com.

• 2-Acessando pela 1a vez

Sempre que você abrir o Portugol Studio, você encontrará na tela inicial, as opções para "Programar", "Ajuda", "Vídeo aulas" ou "Bibliotecas". Além disto, você terá acesso a vários exemplos de códigos, super úteis para se inspirar nas soluções. Explore!



1.3 Link para Download do Portugol Studio

Link para Download do Portugol Studio

Click <https://portugol.dev/> link to open resource.

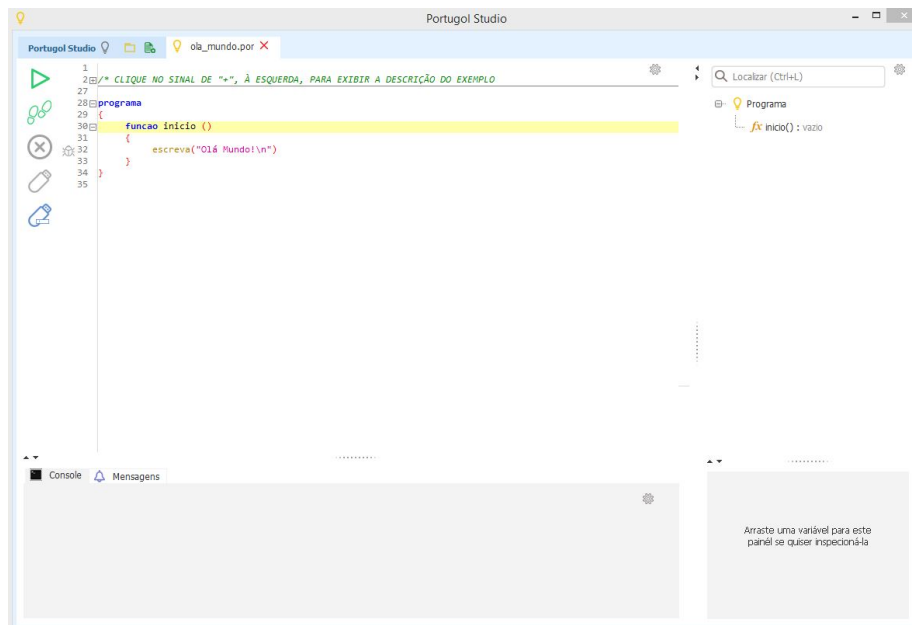
1.4 Meu primeiro programa

Para fazer o seu programa, na tela inicial clique em "Programar". Um arquivo novo chamado "Sem título" é criado. Dentro dele, o Portugol já insere um trecho de código (ver abaixo) com a estrutura básica de um programa.

```
1 programa
2 {
3     funcao inicio()
4     {
5
6     }
7 }
```

Como este arquivo não está mais vazio (afinal, o próprio programa o modificou), o Portugol perguntará se você deseja salvar as alterações quando sair dele. Caso não tenha acrescentado nada ao código, você não precisa salvá-lo (neste caso, selecione a opção "Não").

Quando fizer os exercícios e desejar, de fato, executar um programa, clicando no botão em forma de triângulo na cor verde Ícone verde., no canto superior esquerdo da tela.



1.5 Vídeo de introdução ao Portugol Studio

Caro aluno, recomendamos que você assista este vídeo após ter instalado o software Portugol Studio em seu computador. <https://www.youtube.com/watch?v=8njsRMvongk>

1.6 Variáveis

Ah... as variáveis! Esse nome não te parece estranho? Que bom! Mas se você nunca ouviu falar nessa palavra, ou é porque não frequentava as aulas de matemática ou porque não estava no planeta terra durante as aulas. Brincadeiras a parte, as variáveis são exatamente o que elas significam na língua portuguesa: variam!

As variáveis, portanto, são valores variáveis. Uma hora podem representar algo, e mais tarde, representam outra coisa.

Voltando a matemática, a variável mais famosa é o X. Vamos puxar um pouco na nossa memória uma equação bem simples:

$$x + 5 = 7$$

E aí, te pergunto: qual é o valor de X? Bom, resolvendo a equação, temos...

$$x + 5 = 7$$

$x = 7 - 5$ (lembre da regra, o que soma de um lado, passa para o outro subtraindo...)

$$x = 2$$

Temos que nesta equação o valor de x é igual a 2.

Agora vamos a outra equação (e prometemos ser a última por enquanto):

$$2x + 10 = x - 3$$

Ih, não se desespere, a resolução é mais simples que você imagina:

$2x + 10 = x - 3$ (1° vamos separar os x de um lado, e os números isolados de outro, ok?)

$$2x - x = -10 - 3 \text{ (agora vamos resolver)}$$

$$x = -13$$

Aqui, temos agora que x vale -13.

Bom, como assim? Antes x valia 2 e agora vale -13?? Sim! Afinal, x é uma variável. Ou seja, x não tem valor fixo e definido. A cada momento ele tem um valor diferente.

Seguindo a mesma ideia, as variáveis em um programa de computador servem para armazenar valores variados. Vamos ver um exemplo de um programa com uma variável:

```
1 programa
2 {
3     funcao inicio ()
4     {
5         inteiro x
6         x = 2
7         x = 5
8     }
9 }
```

Aqui, conforme apresentado na figura acima, criamos uma variável `x` cujo valor a ser armazenado deve ser um número inteiro, e por isso a criamos como "inteiro `x`". Na sequência, atribuímos o valor 2 e depois o valor 5. Ou seja, na linha 6, `x` vale 2; mas na linha 7, `x` passou a valer 5.

Além de números inteiros, podemos criar variáveis para outros tipos de informações:

* Números quebrados ou não exatos, como 1,5 (um e meio) - porém, atenção! Os números quebrados na programação são apresentados com `.` (ponto final) ao invés de `,` (vírgula).

real nome_variavel2

* Valor verdadeiro ou falso

logico nome_variavel3

* Letra ou palavras

caracter nome_variavel

Vamos ver um exemplo com todas essas possibilidades de variáveis?

```
1 programa
2 {
3     funcao inicio ()
4     {
5         inteiro idade
6         real peso
7         cadeia nome
8         logico sabenadar
9
10        idade = 22
11        peso = 68.5
12        nome = "Júlia"
13        sabenadar = verdadeiro
14    }
15 }
```

Criamos agora um programa com 4 variáveis:

- `idade`: para armazenar a idade de uma pessoa, que é sempre um número inteiro
- `peso`: para armazenar a quantidade de kilogramas do peso de uma pessoa, que pode ter kilos e gramas
- `nome`: para armazenar uma ou mais palavras que representam o nome (atenção: você pode usar espaços em branco para separar palavras)
- `sabenadar`: para representar se a pessoa sabe ou não nadar

Após, atribuímos valores para as variáveis. Como pode ser visto na figura, a variável `idade` recebe o valor 22, a variável `peso` recebe 68.5, a variável `nome` recebe Júlia (entre aspas duplas, porque é do tipo cadeia) e a variável `sabenadar` receber o valor verdadeiro (não está entre aspas porque não é uma cadeia, mas sim um dos valores possíveis de uma variável do tipo lógico, sendo estes verdadeiro ou falso).

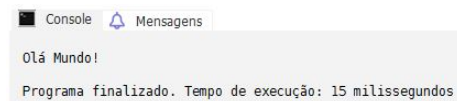
1.7 Saída de Dados

A saída de dados permite que a informação que está dentro do programa seja apresentada para o usuário, ou seja, que possamos mostrar algo que está interno. Para fazer isso no Portugol Studio, temos que utilizar o comando **escreva**.

Vamos ver o exemplo mais clássico da programação, o famoso Hello World, ou aqui no Brasil, o Olá Mundo!

```
programa
{
    funcao inicio ()
    {
        escreva("Olá Mundo!\n")
    }
}
```

Para escrever a mensagem, utilizamos a função escreva acrescida da mensagem desejada escrita entre aspas. Como resultado, teremos a tela abaixo:



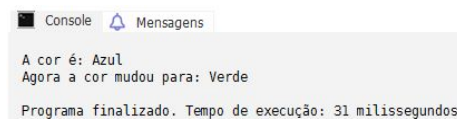
A imagem apresenta o resultado da execução do programa, na aba Console. A aba Console e a aba Mensagens aparecem na parte inferior do Portugol Studio.

E se quisermos exibir o valor de uma variável? Também podemos! Vamos usar nesse próximo exemplo:

```
programa
{
    funcao inicio ()
    {
        cadeia cor
        cor = "Azul"
        escreva("A cor é: ", cor, "\n")
        cor = "Verde"
        escreva("Agora a cor mudou para: ", cor, "\n")
    }
}
```

A imagem apresenta um programa que inicia com a declaração de uma variável do tipo cadeia e que se chama cor. Na sequência é atribuído o valor Azul, em formato cadeia, para a variável cor. Na sequência é impresso, com a função escreva, a mensagem "A cor é: ", concatenada por vírgula com a variável cor, e com o texto "\n", usado para quebrar a linha. Na sequência a variável cor recebe o valor "Verde", e é impressa a mensagem "Agora a cor mudou para: ", concatenado com vírgula com a variável cor e com o texto "\n".

E o resultado será:



A imagem apresenta a execução do programa criado acima. Logo, na aba Console é impresso "A cor é: Azul" e na linha abaixo "Agora a cor mudou para: Verde". Ainda, sempre que um programa é executado é exibida uma mensagem indicando que ele foi finalizado e o seu tempo de execução.

E aí, viu agora como a variável funciona? Uma hora tivemos ela com o valor Azul e logo depois como Verde. Para visualizarmos essa mudança de fora do programa, utilizamos o comando escreva, que exibe mensagens na tela.

1.8 Entrada de Dados

Mas... o que é uma entrada de dados? Bom, todo programa de computador possui uma entrada de dados, embora nem sempre você perceba. Certamente, você já preencheu algum formulário com seu nome, selecionou a cidade em que reside, ou clicou em algum botão em seu computador. Pois bem, quando o computador pede cada uma dessas informações, foi a forma de entrarmos com dados para dentro do computador.

A entrada de dados é quando uma informação externa do programa é repassada para dentro do programa. Ou seja, é a forma do programa de computador receber uma informação.

Como já dissemos, um programa pode pedir para você informar dados pessoais, selecionar uma opção ou até mesmo permitir que você desenhe uma figura. Cada uma dessas informações é obtida através de um comando da linguagem de programação que utilizamos. No caso do Portugol Studio, ela é obtida através do comando leia. Quando executamos esse comando, o computador que roda o programa sabe que deve obter uma informação que irá para dentro do programa através de uma variável. Por isso, é muito importante que você crie as variáveis com nomes compatíveis a informação desejada.

Vamos ver alguns exemplos:

1) Suponha que você precise pedir o nome de uma pessoa, você deve criar um programa assim:

```
programa
{
    funcao inicio ()
    {
        cadeia nome

        escreva("Digite seu nome: ")
        leia(nome)
    }
}
```

A imagem apresenta um programa que inicia com a declaração da variável nome, do tipo cadeia. Após, com a função escreva, é impressa a mensagem "Digite seu nome:". E, por fim, é feita a leitura do que o usuário digitou para nome, com a função leia e entre parênteses a variável que está sendo lida, no caso a variável nome.

2) Suponha que você precise pedir a idade de uma pessoa, você deve criar um programa assim:

```
programa
{
    funcao inicio ()
    {
        inteiro idade

        escreva("Digite sua idade: ")
        leia(idade)

        escreva("A idade digitad foi: ", idade, "\n")
    }
}
```

A imagem apresenta um programa que inicia com a declaração da variável `idade` do tipo `inteiro`. Na sequência é impresso com a função `escreva` a mensagem "Digite sua idade: ". Após isso, é realizada a leitura da variável `idade` utilizando a função `leia`. Ao final do programa é impressa a mensagem "A idade digitada foi:" concatenada com vírgula com a variável `idade` e com o texto `"\n"`, usando a função `escreva`.

Veja que neste segundo exemplo, além de pedirmos a idade, já a exibimos utilizando o comando `escreva`. Para escrever a variável, basta escrever o nome dela, sem utilizar as aspas.

1.8.1 Exemplo: Número Digitado

Exemplo: Utilizando o Portugol Studio, escreva um programa que peça ao usuário para que informe um número inteiro e, na sequência, exiba o número digitado.

Para resolver este problema, vamos começar identificando quais variáveis são necessárias nesse programa? Veja que o enunciado pede apenas um número inteiro. Por isso, vamos criar a variável **"numero"** do tipo **inteiro**.

Depois, veja que o enunciado diz que devemos pedir ao usuário para que informe esse número. Para que o usuário saiba o que deve ser digitado, vamos escrever uma mensagem na tela: **"Digite um número inteiro: "**. E, na sequência, já iremos ler este número do teclado usando a função `leia(numero)`.

Por fim, vamos escrever o número digitado, através do comando `escreva`.

Uma solução possível para este enunciado é apresentada abaixo:

```
1 programa
2 {
3     funcao inicio ()
4     {
5         inteiro numero
6
7         escreva("Digite um número inteiro: ")
8         leia(numero)
9
10        escreva("O número digitado foi: ", numero, "\n")
11    }
12 }
```

A imagem apresenta um programa que inicia com a declaração da variável `numero` do tipo `inteiro`. Na sequência, é impresso o texto "Digite um número inteiro:", com a função `escreva`; e é realizada a leitura do valor digitado pelo usuário para a variável `numero` com a função `leia`. No final, usando a função `escreva`, é impressa a mensagem "O número digitado foi:" concatenada com vírgula com a variável `numero` e com o texto `"\n"`.

Fácil né? Agora, tente fazer um programa similar que peça para informar uma nota de uma prova. Lembre-se que a nota de prova é composta por número quebrado.

1.8.2 Exemplo: Nome

Exemplo: Utilizando o Portugol Studio, escreva um programa que peça ao usuário para que informe o nome de uma pessoa e, na sequência, exiba este nome.

Este exercício é muito parecido com o exemplo anterior. A principal diferença é quanto ao tipo de variável que iremos usar para guardar esse nome. Por isso, é muito importante começar a resolução de um programa identificando as variáveis necessárias. Nesse caso, precisaremos de uma variável **"nome"** do tipo **cadeia**.

Depois disso, basta seguir a mesma lógica do exemplo anterior: solicite ao usuário a informação necessária (um nome), leia esta informação do teclado, e então exiba ela na tela.

Mais uma vez, uma solução possível para este enunciado é apresentada abaixo:

```
1 programa
2 {
3     funcao inicio ()
4     {
5         cadeia nome
6
7         escreva("Digite seu nome: ")
8         leia(nome)
9
10        escreva("Seu nome é: ", nome, "\n")
11    }
12 }
```

A imagem apresenta um programa que inicia com a declaração da variável nome do tipo cadeia. Na sequência é impresso o texto "Digite seu nome", com a função escreva; e realizada a leitura do valor digitado pelo usuário para a variável nome, usando a função leia. No final é impresso o texto "Seu nome é:" concatenado com vírgula com a variável nome e com o texto "\n".

Vamos ousar mais um pouco? Como ficaria um programa que você tenha que pedir o nome e o sobrenome de uma pessoa separadamente e então exibir na tela em uma única mensagem / linha? Vamos lá, tente! Sabemos que você consegue!

1.9 Operadores Aritméticos

Os operadores aritméticos são aqueles que permitem realizar as operações básicas da matemática. Vamos lá, eu sei que você lembra deles ;-)

Soma: permite acumular dois os mais valores. Na programação, ele é representado pelo símbolo +. Exemplo:

```
inteiro valor1, valor2
valor1 = 2
valor2 = 5
resultado = valor1 + valor2
```

Subtração: permite remover dois os mais valores. Na programação, ele é representado pelo símbolo `-`. Exemplo:

```
inteiro valor1, valor2
valor1 = 2
valor2 = 5
resultado = valor1 - valor2
```

Multiplicação: permite multiplicar dois os mais valores. Na programação, ele é representado pelo símbolo `*`. Exemplo:

```
inteiro valor1, valor2
valor1 = 2
valor2 = 5
resultado = valor1 * valor2
```

Divisão: a divisão é ser compreendida por duas operações, a divisão e o resto. Ou seja, na programação, é possível obter o resultado dessas duas operações, tal como em um cálculo normal. Para realizar a divisão, utilizamos o símbolo `/`. Se a divisão envolve números inteiros, ele retornará o valor inteiro da divisão. Se for número real, ele retornará o valor da divisão com as casas decimais. Já para obter o resto, utilizamos o símbolo `%`. Lembrando que o resto sempre se dá em relação a uma divisão inteira. Exemplo:

```
inteiro valor1, valor2
valor1 = 2
valor2 = 5
divisao = valor2 / valor1
resto = valor2
```

Ainda, há de se observar a prioridade de operadores. Tal como na matemática tradicional, na programação são respeitadas as mesmas prioridades. Veja os exemplos:

Exemplo 1: $3 + 7 * 2 \Rightarrow 3 + 14 \Rightarrow 17$

Exemplo 2: $(3 + 7) * 2 \Rightarrow 10 * 2 \Rightarrow 20$

Ou seja, realizamos sempre o que se está entre parênteses por primeiro. Depois, as multiplicações e divisões. E por fim, as somas e subtrações. No caso de mais de uma operação do mesmo tipo, a prioridade é da esquerda para direita.

1.9.1 Exemplo: Quatro Operadores

Exemplo: Utilizando o Portugol Studio, escreva um programa que peça dois números e então exiba o resultado das quatro operações aritméticas básicas entre esses números.

Bom, agora começaremos a ter enunciados um pouco mais complexos... Vamos começar pelas variáveis. Quais que você consegue identificar? Duas delas são fáceis, né? Afinal, o enunciado pede para que sejam informados dois números. Ok, dois números... mas seriam números inteiros ou quebrados (reais)? Neste caso, o enunciado não deixa claro, por isso você pode optar. Porém, vamos dar uma dica: um número inteiro pode ser representado por um número quebrado, ex: 5 inteiro é 5.0 real. Mas e ao contrário? O número 5.5 real pode

ser representado como inteiro? Não, né? Ou será 5, ou será 6? Dessa forma, na dúvida, crie como real e amplie suas possibilidades. As variáveis se chamarão **a** e **b**.

Além desses dois números, note que o enunciado fala que você deverá calcular o resultado das quatro operações aritméticas. Você pode calcular e já exibir na tela, mas o mais fácil é calcular e guardar essa informação. Então também iremos criar mais quatro variáveis reais (já que criamos os números de entrada como reais): **soma**, **sub**, **mult**, **div**.

A partir daí, basta você fazer aquilo que já está expert: peça ao usuário os valores e armazene em a e b. Depois disso, vem o desafio: calcular as operações.

A primeira delas é a soma. Para realizá-la, teremos que pensar de forma inversa ao que o nosso cérebro está acostumado:

Cérebro normal: $a + b = \text{resultado da soma}$

Como o computador pensa: $\text{soma} = a + b$ Isso porque o computador entende que primeiro devemos somar os valores para depois armazená-lo na variável que está à esquerda. Ou seja, ele fará $a + b$ e o resultado disso será enviado a variável **soma**.

Da mesma forma, faremos com as demais operações:

$\text{sub} = a - b$

$\text{mult} = a * b$

$\text{div} = a / b$

Ao final, basta mostrar os resultados na tela!

```
1 programa
2 {
3     funcao inicio()
4     {
5         real a, b, soma, sub, mult, div
6
7         escreva("Digite o primeiro número: ")
8         leia(a)
9
10        escreva("Digite o segundo número: ")
11        leia(b)
12
13        soma = a + b // Soma os dois valores
14        sub = a - b // Subtrai os dois valores
15        mult = a * b // Multiplica os dois valores
16        div = a / b // Divide os dois valores
17
18        escreva("\nA soma dos números é igual a: ", soma) // Exibe o resultado da soma
19        escreva("\nA subtração dos números é igual a: ", sub) // Exibe o resultado da subtração
20        escreva("\nA multiplicação dos números é igual a: ", mult) // Exibe o resultado da multiplicação
21        escreva("\nA divisão dos números é igual a: ", div, "\n") // Exibe o resultado da divisão
22    }
23 }
```

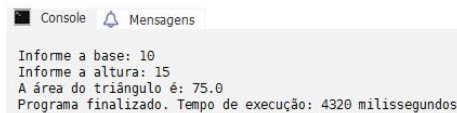
2 Vídeo sobre Entrada e Saída de Dados, Variáveis e Operadores Aritméticos no Portugol Studio

Caro aluno, recomendamos que você leia os materiais anteriores sobre entrada e saída de dados, variáveis e operadores aritméticos antes de visualizar este vídeo.

<https://www.youtube.com/watch?v=5hkYdnhCsn8>

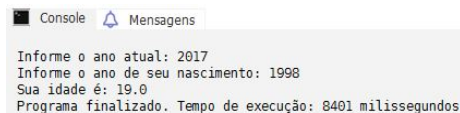
2.1 Praticando um pouco...

- **Exercício 1:** Utilizando o Portugol Studio, escreva um programa que leia o nome de uma pessoa, sua idade e escreva o nome da pessoa e a idade que ela terá daqui 5 anos.
- **Exercício 2:** Imagine que seu professor deu 2 provas e 1 trabalho durante o semestre letivo. A primeira prova teve peso 2. A segunda prova teve peso 1. E o trabalho teve peso 4. Agora que você é um programador, você deseja criar um programa que calcule a sua média nesse semestre. Como seria este programa no Portugol Studio?
- **Exercício 3:** No Portugol Studio, escreva um programa que leia 2 valores: lado e altura. Após, calcule a área do triângulo, sabendo que a fórmula é $\text{área} = (\text{base} * \text{altura}) / 2$



```
Console Mensagens
Informe a base: 10
Informe a altura: 15
A área do triângulo é: 75.0
Programa finalizado. Tempo de execução: 4320 milissegundos
```

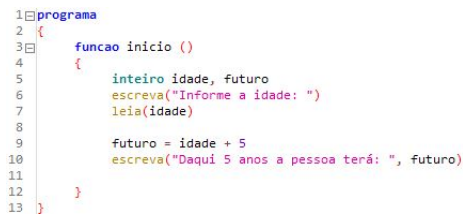
- **Exercício 4:** No Portugol Studio, escreva um programa que peça o ano atual e o ano de seu nascimento. A partir dessas informações, calcule a provável idade da pessoa e exiba na tela



```
Console Mensagens
Informe o ano atual: 2017
Informe o ano de seu nascimento: 1998
Sua idade é: 19.0
Programa finalizado. Tempo de execução: 8401 milissegundos
```

Gabarito

- **Exercício 1: Idade daqui 5 anos**



```
1 programa
2 {
3     funcao inicio ()
4     {
5         inteiro idade, futuro
6         escreva("Informe a idade: ")
7         leia(idade)
8
9         futuro = idade + 5
10        escreva("Daqui 5 anos a pessoa terá: ", futuro)
11    }
12 }
13 }
```

- **Exercício 2: Média ponderada**

```
1 programa
2 {
3     funcao inicio ()
4     {
5         real prova1, prova2, trabalho, media
6         escreva("Informe a nota da prova 1: ")
7         leia(prova1)
8         escreva("Informe a nota da prova 2: ")
9         leia(prova2)
10        escreva("Informe a nota do trabalho: ")
11        leia(trabalho)
12
13        media = (prova1*2 + prova2*3 + trabalho*4) / (2+3+4)
14        escreva("Média final: ", media)
15    }
16 }
17 }
```

- **Exercício 3: Área do triângulo**

```
1 programa
2 {
3     funcao inicio ()
4     {
5         real base, altura, area
6         escreva("Informe a base: ")
7         leia(base)
8         escreva("Informe a altura: ")
9         leia(altura)
10
11        area = (base * altura) / 2
12        escreva("A área do triângulo é: ", area)
13    }
14 }
15 }
```

- **Exercício 4: Idade**

```
1 programa
2 {
3     funcao inicio ()
4     {
5         real anoAtual, anoNascimento, idade
6         escreva("Informe o ano atual: ")
7         leia(anoAtual)
8         escreva("Informe o ano de seu nascimento: ")
9         leia(anoNascimento)
10
11        idade = anoAtual - anoNascimento
12        escreva("Sua idade é: ", idade)
13    }
14 }
15 }
```

2.2 Solucionando Erros

A medida que vamos desenvolvendo programas, é comum nos depararmos com erros no código. O erro pode ocorrer em virtude de um comando mal escrito (sintaxe) ou de raciocínio lógico (semântico). Um link muito legal que explica essa diferença é:

Nos materiais a seguir, daremos algumas dicas de como identificar e resolver alguns problemas de sintaxe e semântica. Para a sintaxe, iremos compreender as mensagens de erro da ferramenta Portugol Studio e como solucioná-las. Para a semântica, vamos ver os testes de mesa, a partir do material criado pelo prof. Edson Pimentel.

2.2.1 Mensagens de Erro

As mensagens de erro, embora odiadas pelos programadores, são as nossas grandes aliadas na construção de programas. Para entender o processo de criação de um programa, é necessário compreender os passos realizados pelo computador.

1º) O programador deve criar o código-fonte

2º) Quando finalizado, realizamos a compilação do programa. Na compilação, é verificado se o programa está escrito corretamente (sem erros de digitação) e se ele funciona (ex: não colocamos uma letra em uma variável inteira).

3º) Caso tudo esteja correto, ele será executado.

O Portugol Studio detecta boa parte dos erros cometidos, visando auxiliar o programador e permitir o funcionamento correto do programa. Vamos ver alguns exemplos de mensagens de erros, ok?

1ª) Quando a variável foi escrita de forma incorreta / diferente de como ela foi declarada ou quando ela não é declarada.

Veja na imagem abaixo, que a variável **nome** foi criada, entretanto quando fomos utilizá-la, a chamamos de **name**. A ferramenta não sabe detectar que nos equivocamos na digitação, pois pode ser que tenhamos feito propositalmente (que queríamos duas variáveis). Mas, veja que a mensagem auxilia na identificação do problema para que possamos corrigí-lo. Aqui, há duas mensagens: uma indicando que a variável não foi declarada (e isso ocorre na linha 36); e uma que informa que a variável nome não foi inicializada, ou seja, nada foi atribuída a ela durante o programa.

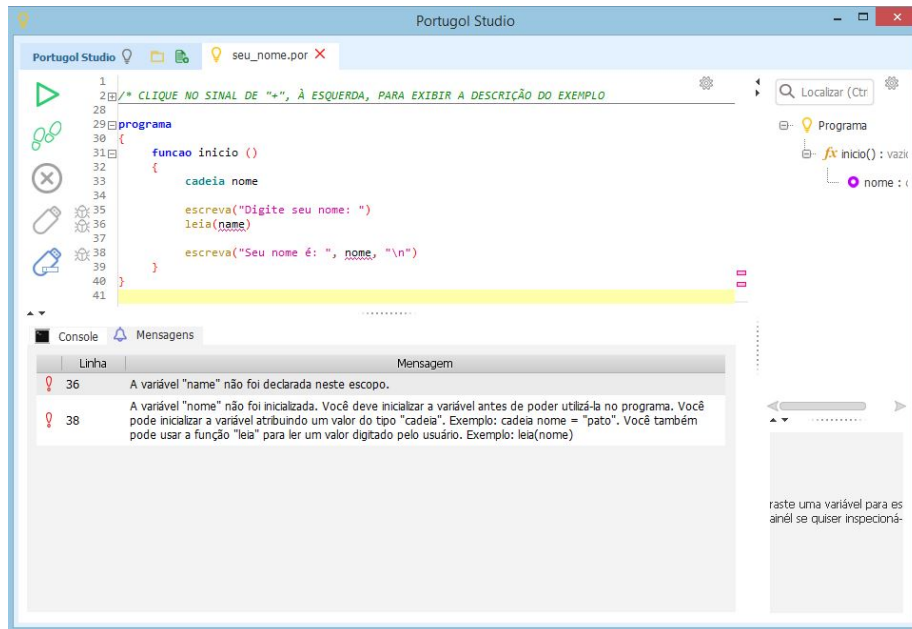


Figura 1: A imagem apresenta o código e os erros descritos acima.

2ª) Quando um comando foi escrito incorretamente.

Há vezes que erramos na digitação de um comando. Quando isto ocorre, o comando não é identificado pela ferramenta e por isso ela nos avisa. Cabe a nós, fazer a devida correção na escrita.

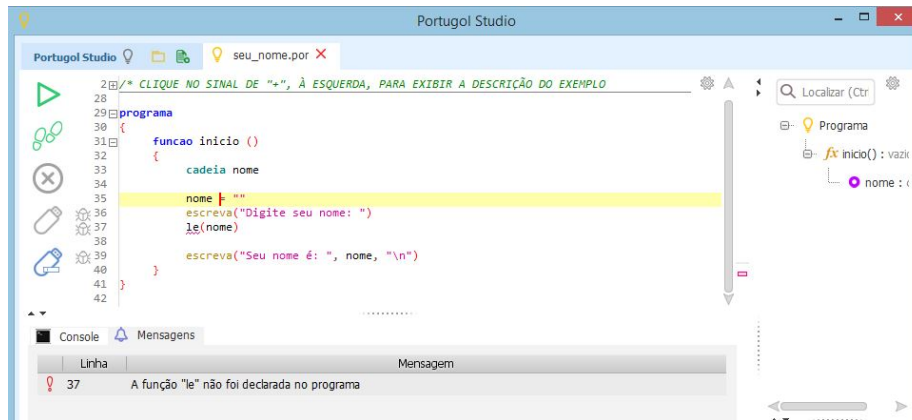


Figura 2: Na imagem é apresentado um programa em que a função `le` é escrita apenas como `le`. O programa indica que há um erro, na aba inferior `Mensagens`. O erro indicado que a função `le` não foi declarada no programa.

3ª) Confusão entre linguagens de programação.

Para quem está aprendendo a programar, pode encontrar diversas "meta-linguagens" de Portugol. Ou seja, cada professor cria sua linguagem para ensinar os seus alunos. Existem professores que para o sinal de atribuição utilizam o símbolo «-», outros uma seta, enquanto aqui no Portugol Studio, utilizamos o "=". Por isso, não fique chateado se você trocar os símbolos ou comandos quando estiver programando. É super normal.

No caso abaixo, o programador se equivocou ao realizar a atribuição na variável `nome`. Note que a mensagem informa que este tipo de expressão lógica não faz sentido no código. Também são exibidas outras duas mensagens: uma referente a inicialização da variável e outra sobre a incompatibilidade da variável. Veja, portanto, que as três mensagens se referem a linha 35, mas em nenhum momento ela diz claramente que "você se confundiu ao escrever o comando de atribuição". Infelizmente não há como detectar todos os equívocos de um cérebro humano. A mensagem te ajudará a localizar um erro, mas a correção caberá a você.

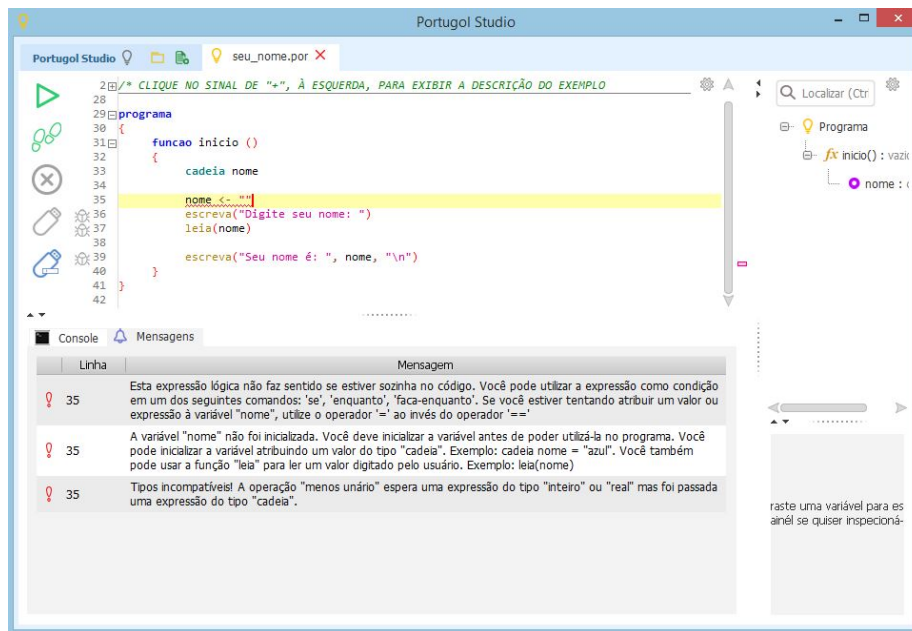


Figura 3: A imagem apresenta o código e os erros descritos acima.

4ª) Variável de um tipo sendo usada para outro.

Este erro é bem comum e facilmente identificado pela ferramenta. Veja que a mensagem indica a incompatibilidade entre valor e variável. Agora é só corrigir.

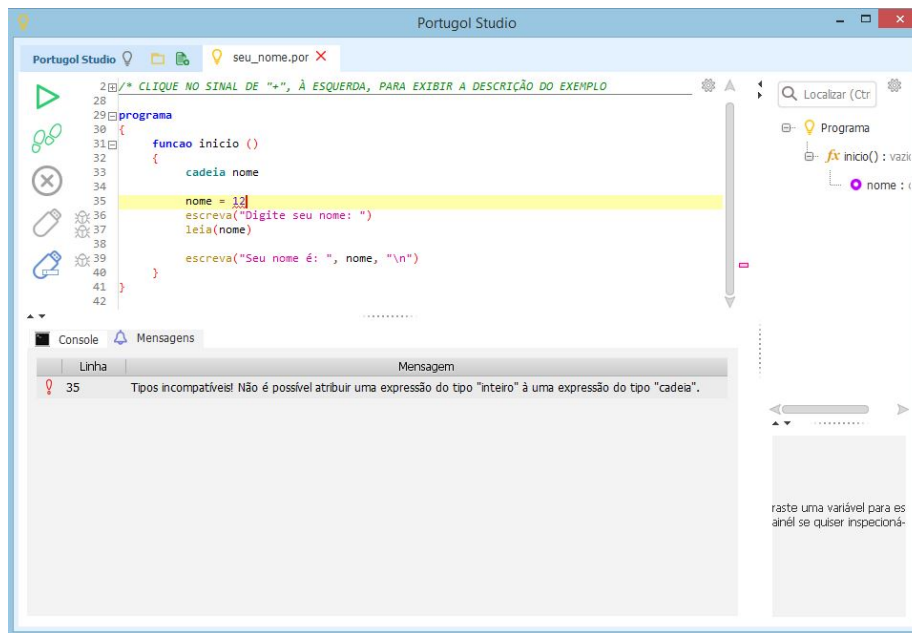


Figura 4: A imagem apresenta um código em que foi declarada uma variável do tipo cadeia e é atribuído a ela um valor inteiro.

2.2.2 Testes de Mesa

Click <https://pt.slideshare.net/henriquecarmona/aula-4-teste-de-mesa> link to open resource.