



Desenvolvimento de Sistemas

Validação e testes: teste de mesa e depuração de algoritmo

Teste de mesa

O teste de mesa é uma técnica de execução manual das linhas de um código com o objetivo de validá-lo, acompanhando a execução de um algoritmo passo a passo ou instrução a instrução. Assim é possível encontrar erros ou validar a lógica idealizada para determinado algoritmo.

Verificar o conteúdo das variáveis dispostas no algoritmo e acompanhar as instruções dadas facilitarão seu acompanhamento. O livro “Técnicas de Programação” (LEITE, 2020) recomenda alguns passos básicos para um bom procedimento de teste de mesa e descreve-os com a seguinte disposição:

1. Elaborar uma tabela onde cada coluna se refere a cada variável envolvida.
2. Executar os passos previstos no algoritmo.
3. Verificar se os resultados obtidos são coerentes com os previstos.
4. Encerrar o teste após um número razoável de resultados corretos obtidos.

Agora é a hora de colocar esses passos em prática!

O primeiro algoritmo é simples. Você escreverá dois valores e mostrará suas saídas aplicando as quatro operações básicas da matemática: adição, subtração, multiplicação e divisão.



N.º da linha	Algoritmo
3	variáveis declaradas (a, b, soma, subtração, divisão, multiplicação)
4	escreva ("Digite o valor de a: ")
5	leia (a)
6	escreva ("Digite o valor de b: ")
7	leia (b)
8	$soma = a + b$
9	leia (soma)
10	$subtração = a - b$
11	leia(subtração)
12	$divisão = a / b$
13	leia(divisão)
14	$multiplicação = a * b$
15	leia(multiplicação)



1.º Passo: neste primeiro passo, verifique e identifique todas as variáveis que estão no algoritmo. As variáveis encontradas são: a, b, soma, subtração, divisão e multiplicação. As variáveis "a" e "b" receberão valores e as demais variáveis receberão suas respectivas operações matemáticas.



2.º Passo: após a identificação das variáveis, será criada uma tabela contendo todas as variáveis declaradas, o passo a ser executado e a linha do programa.



Passo	Linha	a	b	soma	subtração	divisão	multiplicação
-	-	-	-	-	-	-	-



3.º Passo: com as variáveis devidamente identificadas, será necessário incluir seus valores, gerando uma nova tabela com seus respectivos valores. Nessa tabela, a coluna “Passo” receberá “Início”, mostrando exatamente o começo do programa; a coluna “Linha” mostrará a linha que foi executada; as variáveis “a” e “b” receberão os valores desejados; e as variáveis em forma de operações matemáticas armazenarão os resultados das operações.

Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-



4.º Passo: a partir deste passo, todas as linhas serão percorridas, mostrando uma tabela com a linha e suas respectivas informações.

Linha 3

Ao executar a linha 3, todas as variáveis são declaradas. Até o momento, elas não contêm nenhum valor para ser agregado.

Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-

Linha 4



O programa solicitará que seja digitado um valor para a variável “a”. Como exemplo, o valor escolhido será igual a 20.

Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-
2	4	20	-	-	-	-	-

Linha 5

A seguir, após ser executada a linha 4, a variável “a” já estará armazenando o valor 20, que poderá ser visualizado por meio do comando “leia o seu valor”.

Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-
2	4	20	-	-	-	-	-
3	5	20	-	-	-	-	-

Linha 6

Com a variável “a” preenchida, o programa executará a linha 6 e solicitará que seja digitado o valor de “b”, ao qual será atribuído 5 como valor da variável.

Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-
2	4	20	-	-	-	-	-
3	5	20	-	-	-	-	-
4	6	-	5	-	-	-	-

Linha 7

Neste momento, após ser executada a linha 7, a variável “b” já estará armazenando o valor 5, que poderá ser visualizado por meio do comando “leia o seu valor”.

Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-
2	4	20	-	-	-	-	-
3	5	20	-	-	-	-	-
4	6	-	5	-	-	-	-
5	7	-	5	-	-	-	-

Linha 8



A variável “soma” está recebendo duas variáveis, respectivamente “a” e “b”, às quais foram atribuídos anteriormente os valores 20 e 5. Para que seja realizada a soma dos valores, é aplicado o operador “+” da matemática, que realizará efetivamente a soma dos valores das duas variáveis.

Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-
2	4	20	-	-	-	-	-
3	5	20	-	-	-	-	-
4	6	-	5	-	-	-	-
5	7	-	5	-	-	-	-
6	8	-	-	a + b	-	-	-

Linha 9

Após a variável “soma” receber as duas variáveis com a atribuição de adição, o resultado dessa adição é apresentado na linha 9 com o valor de 25, que é a soma de “a” mais “b”.



Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-
2	4	20	-	-	-	-	-
3	5	20	-	-	-	-	-
4	6	-	5	-	-	-	-
5	7	-	5	-	-	-	-
6	8	-	-	a + b	-	-	-
7	9	-	-	25	-	-	-

Linha 10

A variável “subtração” está recebendo duas variáveis, respectivamente “a” e “b”, às quais foram atribuídos anteriormente os valores 20 e 5. Para que seja realizada a subtração dos valores, é aplicado o operador “-” da matemática, que realizará efetivamente a subtração entre os valores das duas variáveis.



Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-
2	4	20	-	-	-	-	-
3	5	20	-	-	-	-	-
4	6	-	5	-	-	-	-
5	7	-	5	-	-	-	-
6	8	-	-	$a + b$	-	-	-
7	9	-	-	25	-	-	-
8	10	-	-	-	$a - b$	-	-

Linha 11

Após a variável “subtração” receber as duas variáveis com a atribuição de subtração, o resultado desse cálculo matemático será apresentado na linha 11 com o valor de 15, que é a diferença entre “a” e “b”.



Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-
2	4	20	-	-	-	-	-
3	5	20	-	-	-	-	-
4	6	-	5	-	-	-	-
5	7	-	5	-	-	-	-
6	8	-	-	$a + b$	-	-	-
7	9	-	-	25	-	-	-
8	10	-	-	-	$a - b$	-	-
9	11	-	-	-	15	-	-

Linha 12

A variável “divisão” está recebendo duas variáveis, respectivamente “a” e “b”, às quais foram atribuídos anteriormente os valores 20 e 5. Para que seja realizada a divisão dos valores, é aplicado o operador “/” da matemática, que realizará efetivamente a divisão entre os valores das duas variáveis.



Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-
2	4	20	-	-	-	-	-
3	5	20	-	-	-	-	-
4	6	-	5	-	-	-	-
5	7	-	5	-	-	-	-
6	8	-	-	$a + b$	-	-	-
7	9	-	-	25	-	-	-
8	10	-	-	-	$a - b$	-	-
9	11	-	-	-	15	-	-
10	12	-	-	-	-	a / b	-

Linha 13

Após a variável “divisão” receber as duas variáveis com a atribuição de divisão, o resultado desse cálculo matemático será apresentado na linha 13, com o valor de 4, que é o resultado da divisão de “a” por “b”.



Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-
2	4	20	-	-	-	-	-
3	5	20	-	-	-	-	-
4	6	-	5	-	-	-	-
5	7	-	5	-	-	-	-
6	8	-	-	$a + b$	-	-	-
7	9	-	-	25	-	-	-
8	10	-	-	-	$a - b$	-	-
9	11	-	-	-	15	-	-
10	12	-	-	-	-	a / b	-
11	13	-	-	-	-	4	-

Linha 14

A variável “multiplicação” está recebendo duas variáveis, respectivamente “a” e “b”, às quais foram atribuídos anteriormente os valores 20 e 5. Para que seja realizada a multiplicação desses valores, é aplicado o operador “*” da matemática, que realizará efetivamente a multiplicação dos valores das duas variáveis.



Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-
2	5	12	-	-	-	-	-
3	6	12	-	-	-	-	-
4	7	-	3	-	-	-	-
5	8	-	3	-	-	-	-
6	9	-	-	$a + b$	-	-	-
7	10	-	-	15	-	-	-
8	11	-	-	-	$a - b$	-	-
9	12	-	-	-	9	-	-
10	13	-	-	-	-	a / b	-
11	14	-	-	-	-	4	-
12	15	-	-	-	-	-	$a * b$

Linha 15

Após a variável “multiplicação” receber as duas variáveis com a atribuição de multiplicação, o resultado desse cálculo matemático será apresentado na linha 15, com o valor de 100, que é o resultado da multiplicação de “a” por “b”.



Passo	Linha	a	b	soma	subtração	divisão	multiplicação
Início	-	-	-	-	-	-	-
1	3	-	-	-	-	-	-
2	4	20	-	-	-	-	-
3	5	20	-	-	-	-	-
4	6	-	5	-	-	-	-
5	7	-	5	-	-	-	-
6	8	-	-	$a + b$	-	-	-
7	9	-	-	25	-	-	-
8	10	-	-	-	$a - b$	-	-
9	11	-	-	-	15	-	-
10	12	-	-	-	-	a / b	-
11	13	-	-	-	-	4	-
12	14	-	-	-	-	-	$a * b$
13	15	-	-	-	-	-	100

Teste de mesa com Portugol

A representação do algoritmo em Portugol, conhecido também por ser um pseudocódigo, é muito utilizada para a aprendizagem de programação. Segundo Saliba (1992, p. 6): “Esta forma de representação de algoritmo é rica em detalhes, como a definição dos tipos das variáveis usadas no algoritmo, e, por assemelhar-se bastante à forma em que os programas são escritos, encontra muita aceitação”.

Portugol – Depuração

O Portugol ainda conta com um ambiente de desenvolvimento próprio que facilita sua utilização. Uma das grandes vantagens de se utilizar um ambiente já consolidado é que ele conta com um modo de depuração agregado em suas funcionalidades básicas. O modo de depuração permite executar o código passo a passo para descobrir onde o desenvolvedor possa ter cometido um erro em seu algoritmo.

Esse modo de execução permite que seja marcada a linha na qual se quer ver o erro. Após a marcação da linha, um ícone de “inseto” ficará da cor vermelha, como se fosse um “flag”. Ao executar o algoritmo, ele executará as linhas que receberão essa marcação e será possível realizar a depuração de todo o código.



Figura 1 – Funcionalidades básicas do Portugol disponíveis na aba esquerda do ambiente de programação

A imagem mostra a aba esquerda do ambiente de desenvolvimento Portugol contendo suas funcionalidades básicas representadas por ícones: triângulo (iniciar), pegadas (depuração), letra “x” envolvida por um círculo (parar depuração), *pen drive* (salvar), *pen drive* com retângulo (salvar como), pasta (abrir arquivo) e ícone de interrogação (ajuda). Ainda na mesma imagem, há a numeração das linhas nas quais será desenvolvido o algoritmo e o ícone inseto (marcação de *flag*).

No exemplo a seguir, você verá um algoritmo em Portugol ilustrando como é realizar um teste de mesa com uma representação de código. Será abordada a seguinte problemática:

Ao realizar o cadastro para o processo da primeira habilitação em um centro de formação de condutores, o candidato precisa ter idade maior ou igual a 18 anos. Para verificar esse dado, foi criado o seguinte algoritmo em Portugol.

```
1  programa
2  {
3      funcao inicio ()
4      {
5          inteiro idade
6
7          escreva("Informe sua idade: ")
8          leia(idade)
9
10         se (idade >= 18)
11         {
12             escreva("Você é maior de idade, pode solicitar sua CNH!!")
13         }
14         senao
15         {
16             escreva("Você é menor de idade, não pode dirigir ainda!!!")
17         }
18     }
19 }
20 }
21 }
```

Figura 2 – Código desenvolvido usando Portugol

A imagem mostra as linhas do algoritmo proposto contendo a declaração da função “início” e do tipo da variável “idade” como inteiro; os comandos “escreva: ‘Informe sua idade’”, para que o usuário informe sua idade, e “leia” para que o código faça a leitura; a condição proposta, por meio das funções “se (idade >= 18 anos)/escreva ‘Você é maior de idade, pode solicitar a sua CNH!’” e “senão/escreva ‘Você é menor de idade, não pode dirigir ainda!’”.

Para verificar se o algoritmo está realmente correto, o teste de mesa será usado para analisar todas as condições, inclusive a condição que constata se o candidato é maior ou menor de idade. Neste primeiro exemplo, o candidato tem 33 anos de idade.

A variável identificada é a “idade”, que receberá obrigatoriamente um valor inteiro.

Passo	Linha	idade
-	-	-

Linha 7

Na linha 7, o programa solicita que seja informada a idade do candidato, que foi definida como 33 anos no exemplo.

Passo	Linha	idade
Início	-	-
1	7	33

Linha 8

O programa lerá o valor “33”.



Passo	Linha	idade
Início	-	-
1	7	33
2	8	33

Linha 10

Neste momento, o algoritmo executará a condição criada, ou seja, verificará se a idade do candidato é maior ou igual a 18 anos.

Passo	Linha	idade
Início	-	-
1	7	33
2	8	33
3	10	Se (33 >= 18) verdadeiro

Linha 12

Devido ao fato de o resultado da condição ser verdadeiro, ou seja, o candidato tem idade superior ou igual a 18 anos, na linha 18, o resultado será a escrita: “Você é maior de idade, pode solicitar a sua CNH!!”.



Passo	Linha	idade
Início	-	-
1	7	33
2	8	33
3	10	Se (33 >= 18) verdadeiro
4	12	Maior de idade

Neste momento, o programa será encerrado, pois a condição foi cumprida com sucesso.

Veja agora o comportamento do algoritmo caso o candidato tenha idade inferior a 18 anos, por exemplo, 17 anos.

Linha 7

Na linha 7, o programa solicita que seja informada a idade do candidato, definida como 17 anos.

Passo	Linha	idade
Início	-	-
1	7	17

Linha 8

O programa lerá o valor “17”.



Passo	Linha	idade
Início	-	-
1	7	17
2	8	17

Linha 10

Neste momento, o algoritmo executará a condição criada, ou seja, verificará se a idade do candidato é maior ou igual a 18 anos.

Passo	Linha	idade
Início	-	-
1	7	17
2	8	17
3	10	Se (17 >= 18) falso

Linha 17

Como o resultado da condição é falso, ou seja, o candidato é menor de 18 anos, o programa automaticamente passa para a linha 15 e cumpre a condição de “senão”, escrevendo a frase da linha 17: “Você é menor de idade, não pode dirigir ainda”.



Passo	Linha	idade
Início	-	-
1	7	17
2	8	17
3	10	(17 >= 18) Falso
4	17	Menor de idade

Teste de mesa com laço de repetição

O teste de mesa também é importante em algoritmos que necessitam de um laço de repetição, ou seja, comandos presentes em blocos que serão repetidos diversas vezes. Por meio dos laços de repetição, é possível criar programas que percorram *arrays* ou que se repitam até atingir uma determinada condição preestabelecida.

Para testar o laço de repetição, será criado um algoritmo que peça sempre um valor positivo ao ser digitado um valor negativo.

```

1 algoritmo "ContaPositivo"
2 var
3 numero: inteiro
4 inicio
5     repita
6         escreva("Digite um número Positivo: ")
7         leia(numero)
8         ate(numero > 0)
9         escreva("Parabéns, você digitou um número positivo.")
10 fimalgoritmo
11

```

(objetos/figura_3.png)

Figura 3 – Código desenvolvido usando Portugol

A imagem mostra as linhas do algoritmo proposto, contendo: a declaração do algoritmo "ContaPositivo" e do tipo da variável "número" como inteiro; o comando "escreva: 'Digite um número Positivo:'" para que o usuário informe um número positivo e "leia" para que o código faça a

leitura; o laço condicional proposto, por meio das funções “repita” e “até (número > 0)”, para que o algoritmo repita a função até que o valor digitado contemple a condição, ou seja, o número digitado seja maior que zero, nesse caso, é executada a linha seguinte com o comando “escreva: Parabéns você digitou um número positivo”. A seguir, o algoritmo é finalizado.

Para realizar o teste de mesa nesse algoritmo, será usado o mesmo esquema estabelecido nos exemplos anteriores.

É necessário identificar a variável. Nesse programa, a variável encontrada é chamada “numero” e receberá obrigatoriamente um valor inteiro.

Passo	Linha	idade
-	-	-

Linha 3

Na linha 3, dentro do escopo de variáveis, é declarada a variável “numero” como inteiro.

Passo	Linha	idade
Início	-	-
1	3	-

Linha 4

Logo após, na linha 4, o programa é iniciado através da palavra reservada “Início”.

Linha 5

Nesta linha, é declarado o laço de repetição através da palavra reservada “repita”, que fará efetivamente com que o laço seja aplicado no programa.

Linha 6

Ao entrar no laço de repetição, o programa exibirá a frase “Digite um número positivo”. O valor fornecido como exemplo será “-5”.

Passo	Linha	idade
Início	-	-
1	6	-5

Linha 7

Por meio da palavra reservada “leia”, o programa fará a leitura do que foi escrito na linha anterior, ou seja, o valor “-5”.



Passo	Linha	idade
Início	-	-
1	6	-5
2	7	-5

Linha 8

Nesta linha, encontra-se a palavra reservada “ate”, que fará o programa verificar se a condição proposta foi atingida, ou seja, se o número digitado é maior que 0 (número positivo).

Passo	Linha	idade
Início	-	-
1	6	-5
2	7	-5
3	8	$(-5 > 0)$ Falso

Para o exemplo proposto, com valor igual a -5, a condição não foi atingida, ou seja, o valor -5 é menor que 0. Então obrigatoriamente o programa retorna para a linha 6.

Linha 6

Ao retornar à linha 6, o programa entra no laço de repetição novamente e escreve a frase “Digite um número positivo”. Esse comportamento será repetido enquanto a condição proposta não for cumprida, supondo que, em dado momento, o usuário fornecerá um valor positivo, por exemplo, “5”.

Passo	Linha	idade
Início	-	-
1	6	5

Linha 7

O programa faz a leitura do que foi escrito na linha anterior, ou seja, o valor 5.

Passo	Linha	idade
Início	-	-
1	6	5
2	7	5

Linha 8

Nesta linha, há a palavra reservada “ate”, que fará o programa verificar se a condição proposta foi atingida, ou seja, se o número digitado é maior que 0 (número positivo).

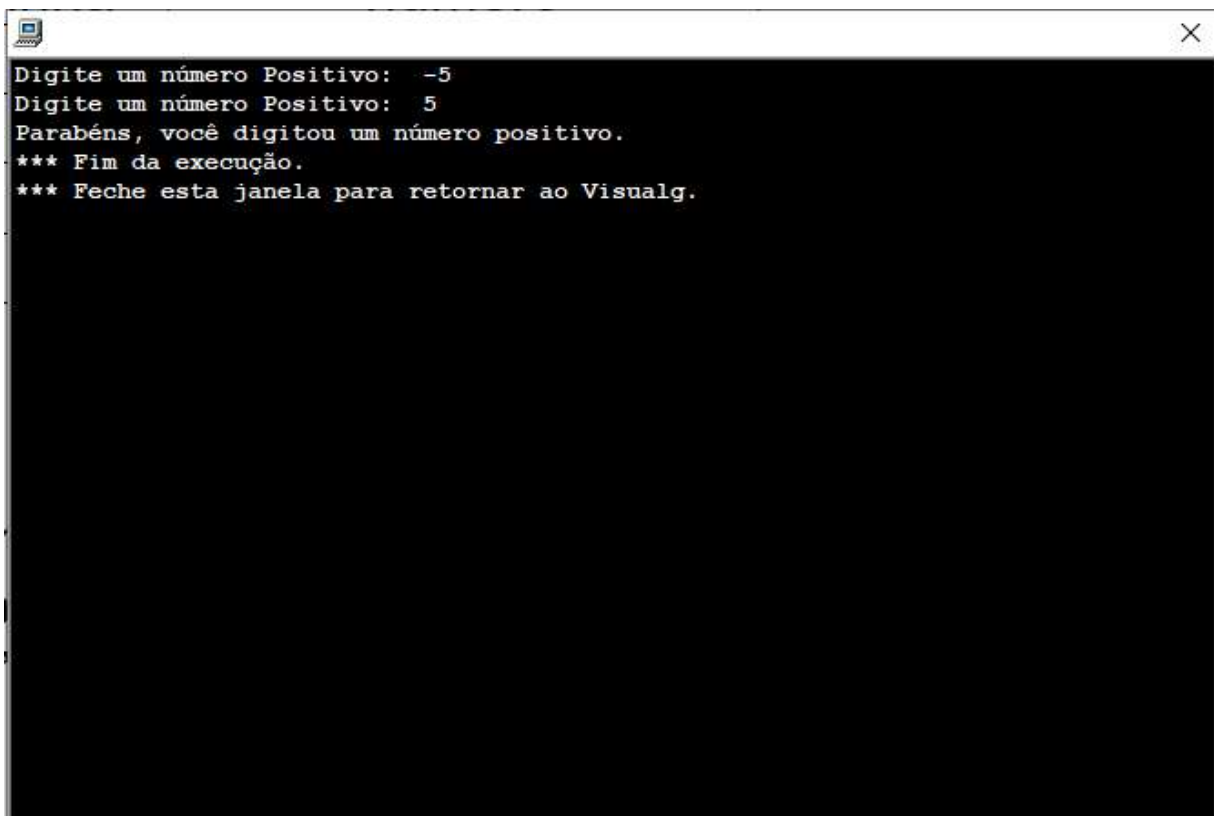


Passo	Linha	idade
Início	-	-
1	6	5
2	7	5
3	8	(5 > 0) Verdadeiro

Linha 9

Agora com o valor fornecido atendendo à condição determinada, o algoritmo escreverá a frase: “Parabéns, você digitou um número positivo”, finalizando o programa.

A figura 4, a seguir, ilustra a execução do exemplo do teste de mesa com laço de repetição.



```
Visual Basic
X
Digite um número Positivo: -5
Digite um número Positivo: 5
Parabéns, você digitou um número positivo.
*** Fim da execução.
*** Feche esta janela para retornar ao Visualg.
```

(objetos/figura_4.png)



Figura 4 – Exemplo do resultado do teste de mesa com laço de repetição

A imagem mostra o resultado do código executado. A primeira linha contém “Digite um número positivo” com o valor digitado “-5”. Na segunda linha, o mesmo texto da linha anterior é repetido, mas com o valor digitado “5”, atendendo à condição verdadeira do laço. Logo após é informado o seguinte texto: “Parabéns, você digitou um número positivo”. Para finalizar, as últimas linhas contêm as seguintes informações: fim da execução e feche esta janela para retornar ao Visualg.

Conforme os exemplos citados, é possível perceber o quanto os testes de mesa contribuem na construção do pensamento de algoritmos, pois ajudam a simular a execução do programa. Dessa forma, o programador pode avaliar o desenvolvimento do código passo a passo e corrigir problemas e/ou imperfeições.