

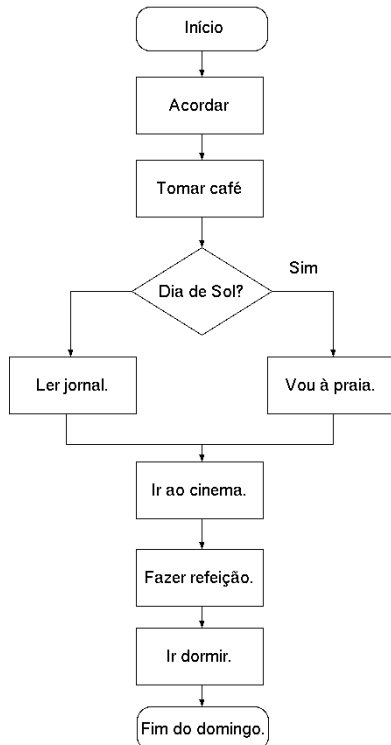
# Ciência da Computação

Prof. Dr. Leandro Alves Neves

Aula 06 apoio

Algoritmos e Programação

Fluxograma para um domingo



# Sumário

- Estratégia para definir um algoritmo
  - Etapas para definir um modelo
  - Algoritmo
  - Codificação
  - Erros
  - Variações de Códigos

# Estratégia para definir um algoritmo

## ■ Problema:

- ❑ 1ª. Leitura do enunciado
- ❑ 2ª. Resumir o objetivo
- ❑ 3ª. Identificar as entradas, o Processamento e as Saídas
- ❑ 4ª. Escrever um algoritmo
- ❑ 5ª. Codificar

# Estratégia para definir um algoritmo

- 1ª Etapa (*Leitura*): Exemplo, Exercício 2 da Lista 03
  - *Elaborar um algoritmo para calcular o reajuste de salário de um funcionário. Considere que o funcionário receberá um reajuste de 15%, caso seu salário seja menor que R\$ 500,00. Se o salário for maior ou igual a R\$ 500,00, mas menor ou igual a R\$ 1.000,00, o reajuste será de 10%. Se o salário for maior que R\$ 1.000,00, o reajuste aplicado será de 5%. Mostrar o salário com reajuste.*
  - 2ª. Resumo: *Calcular e Mostrar o salário com um reajuste*

# Estratégia para definir um algoritmo

- 3ª Etapa: Identificar Entrada(s), Processamento e Saída(s)
  - *Elaborar um algoritmo para calcular o reajuste de salário de um funcionário. Considere que o funcionário receberá um reajuste de 15%, caso seu salário seja menor que R\$ 500,00. Se o salário for maior ou igual a R\$ 500,00, mas menor ou igual a R\$ 1.000,00, o reajuste será de 10%. Se o salário for maior que R\$ 1.000,00, o reajuste aplicado será de 5%. Mostrar o salário com reajuste.*
  - **Identificar Entrada(s):** Uma entrada pode ser um valor fornecido pelo usuário ou gerado pelo próprio algoritmo. A entrada deve ser utilizada no processamento ou na saída do algoritmo. Na saída, o objetivo é informar o usuário ou associar o resultado com elementos do problema, por exemplo.

# Estratégia para definir um algoritmo

- 3ª Etapa: Identificar **Entrada(s)**, Processamento e Saída(s)
  - *Elaborar um algoritmo para calcular o reajuste de salário de um funcionário. Considere que o funcionário receberá um reajuste de 15%, caso seu salário seja menor que R\$ 500,00. Se o salário for maior ou igual a R\$ 500,00, mas menor ou igual a R\$ 1.000,00, o reajuste será de 10%. Se o salário for maior que R\$ 1.000,00, o reajuste aplicado será de 5%. Mostrar o salário com reajuste.*

Entrada(s)	Processamento	Saída(s)
•Salário funcionário: sal_func		

# Estratégia para definir um algoritmo

- 3ª Etapa: Identificar Entrada(s), Processamento e Saída(s)
  - *Elaborar um algoritmo para calcular o reajuste de salário de um funcionário. Considere que o funcionário receberá um reajuste de 15%, caso seu salário seja menor que R\$ 500,00. Se o salário for maior ou igual a R\$ 500,00, mas menor ou igual a R\$ 1.000,00, o reajuste será de 10%. Se o salário for maior que R\$ 1.000,00, o reajuste aplicado será de 5%. Mostrar o salário com reajuste.*

Entrada(s)	Processamento	Saída(s)
• Salário funcionário: sal_func		• salário      funcionário: sal_func • salário func. reajuste: sal_reajuste

# Estratégia para definir um algoritmo

- 3ª Etapa: Identificar Entrada(s), **Processamento** e Saída(s)
  - *Elaborar um algoritmo para calcular o reajuste de salário de um funcionário. Considere que o funcionário receberá um reajuste de 15%, caso seu salário seja menor que R\$ 500,00. Se o salário for maior ou igual a R\$ 500,00, mas menor ou igual a R\$ 1.000,00, o reajuste será de 10%. Se o salário for maior que R\$ 1.000,00, o reajuste aplicado será de 5%. Mostrar o salário com reajuste.*

Entrada(s)	Processamento	Saída(s)
<ul style="list-style-type: none"> <li>• Salário funcionário: <code>sal_func</code></li> </ul>	<ul style="list-style-type: none"> <li>• <b>reajuste 15%: <math>sal\_reajuste = sal\_func + sal\_func * 0.15</math></b></li> <li> </li> <li>se (<code>sal_func &lt; 500</code>)  <math>sal\_reajuste = sal\_func + sal\_func * 0.15</math></li> </ul>	<ul style="list-style-type: none"> <li>• salário funcionário: <code>sal_func</code></li> <li>• salário func. reajuste: <code>sal_reajuste</code></li> </ul>



# Estratégia para definir um algoritmo

- 3ª Etapa: Identificar Entrada(s), **Processamento** e Saída(s)
  - *Elaborar um algoritmo para calcular o reajuste do salário de um funcionário. Considere que o funcionário receberá um reajuste de 15%, caso seu salário seja menor que R\$ 500,00. Se o salário for maior ou igual a R\$ 500,00, mas menor ou igual a R\$ 1.000,00, o reajuste será de 10%. Se o salário for maior que R\$ 1.000,00, o reajuste aplicado será de 5%. Mostrar o salário com reajuste.*

Entrada(s)	Processamento	Saída(s)
<ul style="list-style-type: none"> <li>• Salário funcionário: <code>sal_func</code></li> </ul>	<ul style="list-style-type: none"> <li>• reajuste 15%: <math>sal\_reajuste = sal\_func + sal\_func * 0.15</math>                se (<code>sal_func &lt; 500</code>)                    <math>sal\_reajuste = sal\_func + sal\_func * 0.15</math></li> <li>• reajuste 10%: <math>sal\_reajuste = sal\_func + sal\_func * 0.10</math>                se (<code>sal_func &gt;= 500</code> e <code>sal_func &lt;= 1000</code>)                    <math>sal\_reajuste = sal\_func + sal\_func * 0.10</math></li> </ul>	<ul style="list-style-type: none"> <li>• salário funcionário: <code>sal_func</code></li> <li>• salário func. reajuste: <code>sal_reajuste</code></li> </ul>

# Estratégia para definir um algoritmo

## ■ 3ª Etapa: Identificar Entrada(s), **Processamento** e Saída(s)

- *Elaborar um algoritmo para calcular o reajuste de salário de um funcionário. Considere que o funcionário receberá um reajuste de 15%, caso seu salário seja menor que R\$ 500,00. Se o salário for maior ou igual a R\$ 500,00, mas menor ou igual a R\$ 1.000,00, o reajuste será de 10%. Se o salário for maior que R\$ 1.000,00, o reajuste aplicado será de 5%. Mostrar o salário com reajuste.*

Entrada(s)	Processamento	Saída(s)
<ul style="list-style-type: none"> <li>• Salário funcionário: <code>sal_func</code></li> </ul>	<ul style="list-style-type: none"> <li>• Reajuste 15%: <math>sal\_reajuste = sal\_func + sal\_func * 0.15</math> se (<code>sal_func &lt; 500</code>) <math>sal\_reajuste = sal\_func + sal\_func * 0.15</math></li> <li>• Reajuste 10%: <math>sal\_reajuste = sal\_func + sal\_func * 0.10</math> se (<code>sal_func &gt;= 500</code> e <code>sal_func &lt;= 1000</code>) <math>sal\_reajuste = sal\_func + sal\_func * 0.10</math></li> <li>• Caso contrário (<code>&gt;1000</code>), reajuste 5%: <math>sal\_reajuste = sal\_func + sal\_func * 0.05</math> senão <math>sal\_reajuste = sal\_func + sal\_func * 0.05</math></li> </ul>	<ul style="list-style-type: none"> <li>• salário funcionário: <code>sal_func</code></li> <li>• salário func. reajuste: <code>sal_reajuste</code></li> </ul>

# Estratégia para definir um algoritmo

## ■ 4ª Algoritmo

Entrada(s)

•Salário  
funcionário:  
**sal\_func**

□ programa salario\_funcionario // nome do programa  
início

**real sal\_func, sal\_reajuste;**

escreva (“Digite o salário do funcionário”);

leia (sal\_func);

/\*

instruções

\*/

fim

// comentar linha

/\* comentar bloco \*/

# Estratégia para definir um algoritmo

## ■ 4ª Algoritmo

□ programa salario\_funcionario // nome do programa  
início

```
real sal_func, sal_reajuste;  
escreva ("Digite o salário do funcionário");  
leia (sal_func);
```

//instruções

```
escreva ("O salário do funcionário é: ", sal_func);  
escreva ("O salário com reajuste é: ", sal_reajuste);
```

fim

### Saída(s)

- salário  
funcionário:  
sal\_func
- salário func.  
reajuste:  
sal\_reajuste

# Estratégia para definir um algoritmo

## ■ 4ª Algoritmo (//comentário)

□ programa salario\_funcionario

início

real sal\_func, sal\_reajuste;

escreva (“Digite o salário do funcionário”);

leia (sal\_func);

se (sal\_func < 500)

sal\_reajuste=sal\_func + sal\_func\*0.15;

senão se (sal\_func >= 500 e sal\_func<=1000)

sal\_reajuste=sal\_func + sal\_func\*0.10;

senão

sal\_reajuste=sal\_func + sal\_func\*0.05;

escreva (“O salário do funcionário é: ”, sal\_func);

escreva (“O salário com reajuste é: ”, sal\_reajuste);

fim

### Processamento

- Reajuste 15%:  $sal\_reajuste = sal\_func + sal\_func * 0.15$   
se (sal\_func < 500)  
 $sal\_reajuste = sal\_func + sal\_func * 0.15$
- Reajuste 10%:  $sal\_reajuste = sal\_func + sal\_func * 0.10$   
se (sal\_func >= 500 e sal\_func <= 1000)  
 $sal\_reajuste = sal\_func + sal\_func * 0.10$
- Caso contrário (>1000), reajuste 5%:  
 $sal\_reajuste = sal\_func + sal\_func * 0.05$   
senão  
 $sal\_reajuste = sal\_func + sal\_func * 0.05$

# Estratégia para definir um algoritmo

## Algoritmo x Codificação

□ programa salario\_funcionario

início

real sal\_func, sal\_reajuste;

escreva ("Digite o salário do funcionário");

leia (sal\_func);

se (sal\_func < 500)

sal\_reajuste=sal\_func + sal\_func\*0.15;

senão se (sal\_func >= 500 e sal\_func <= 1000)

sal\_reajuste=sal\_func + sal\_func\*0.10;

senão

sal\_reajuste=sal\_func + sal\_func\*0.05;

escreva ("O salário do funcionário é: ", sal\_func);

escreva ("O salário com reajuste é: ", sal\_reajuste);

fim

salario\_funcionario.cpp

```

2  #include <stdio.h>
3  #include <stdlib.h>
4  //Obrigatório. Função principal: indica o início da execução do programa
5  int main ()
6  {
7      //Declaração de Variáveis Locais.
8      float sal_func, sal_reajuste;
9      //Obrigatório. Comandos para resolução do problema
10     printf("Digite o salário do funcionário: ");
11     scanf("%f",&sal_func);
12     if (sal_func<500)
13     {
14         sal_reajuste=sal_func+sal_func*0.15;
15     }
16     else if (sal_func>=500 && sal_func<=1000)
17     {
18         sal_reajuste=sal_func+sal_func*0.10;
19     }
20     else
21     {
22         sal_reajuste=sal_func+sal_func*0.05;
23     }
24     printf("\nO salário do funcionário é: %.2f", sal_func);
25     printf("\nO salário com reajuste é: %.2f", sal_reajuste);
26     //Opcional. Comando para interromper momentaneamente o programa
27     printf("\n");
28     system("PAUSE");
29     //Retorno ao SO o status do programa
30     return 0;
31 } //Indica o final do programa.

```

# Estratégia para definir um algoritmo

## ■ Codificação: Alguns Exemplos de Erros, usando Dev-C++ 5.11

salario\_funcionario.cpp

```

2  #include <stdio.h>
3  #include <stdlib.h>
4  //Obrigatório. Função principal: indica o início da execução do programa
5  int main ()
6  {
7      //Declaração de Variáveis Locais.
8      float sal_func, sal_reajuste;
9      //Obrigatório. Comandos para resolução do problema
10     printf("Digite o salário do funcionário: ");
11     scanf("%f",&sal_func);
12     if (sal_func<500)
13     {
14         sal_reajuste=sal_func+sal_func*0.15;
15     }
16 }

```

Compilador (12) Recursos Registro do Compilador Depurador Resultados da Busca Fechar

Lin...	Col...	Unidade	Mensagem
8	2	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected initializer before 'float'
10	9	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected constructor, destructor, or type conversion before '(' token
11	8	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected constructor, destructor, or type conversion before '(' token
12	3	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected unqualified-id before 'if'
16	2	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected unqualified-id before 'else'
20	2	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected unqualified-id before 'else'
24	11	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected constructor, destructor, or type conversion before '(' token
25	9	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected constructor, destructor, or type conversion before '(' token
27	10	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected constructor, destructor, or type conversion before '(' token
28	10	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected constructor, destructor, or type conversion before '(' token
30	4	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected unqualified-id before 'return'
31	1	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected declaration before ')' token

# Estratégia para definir um algoritmo

## ■ Codificação: Alguns Exemplos de Erros, usando Dev-C++ 5.11

salario\_funcionario.cpp

```

2  #include <stdio.h>
3  #include <stdlib.h>
4  //Obrigatório. Função principal: indica o início da execução do programa
5  int main ()
6  {
7      //Declaração de Variáveis Locais.
8      float sal_func, sal_reajuste ;
9      //Obrigatório. Comandos para resolução do problema
10     printf("Digite o salário do funcionário: ");
11     scanf("%f",&sal_func);
12     if (sal_func<500)
13     {
14         sal_reajuste=sal_func+sal_func*0.15;

```

Compilador (6)
 Recursos
 Registro do Compilador
 Depurador
 Resultados da Busca
 Fechar

Lin...	Col...	Unidade	Mensagem
		G:\BKP\Users\User\Desktop\Backup\25_02_2011\Le...	In function 'int main()':
10	3	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected initializer before 'printf'
14	4	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] 'sal_reajuste' was not declared in this scope
18	4	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] 'sal_reajuste' was not declared in this scope
22	3	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] 'sal_reajuste' was not declared in this scope
25	46	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] 'sal_reajuste' was not declared in this scope



# Estratégia para definir um algoritmo







- Codificação: Alguns Exemplos de Erros, usando Dev-C++ 5.11

salario\_funcionario.cpp

```

2  #include <stdio.h>
3  #include <stdlib.h>
4  //Obrigatório. Função principal: indica o início da execução do programa
5  int main ()
6  {
7      //Declaração de Variáveis Locais.
8      float sal_func, sal_reajuste;
9      //Obrigatório. Comandos para resolução do problema
10     printf("Digite o salário do funcionário: ");
11     scanf("%f",&sal_func);
12     if (sal_func<500)
13     {
14         sal_reajuste=sal_func+sal_func*0.15;

```

 Compilador (4)
  Recursos
  Registro do Compilador
  Depurador
  Resultados da Busca
  Fechar

Lin...	Col...	Unidade	Mensagem
10	10	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Warning] missing terminating " character
10	3	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] missing terminating " character
		G:\BKP\Users\User\Desktop\Backup\25_02_2011\Le...	In function 'int main()':
11	24	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected ')' before ';' token

# Estratégia para definir um algoritmo

- Codificação: Alguns Exemplos de Erros, usando Dev-C++ 5.11

The screenshot shows the Dev-C++ 5.11 IDE with a C++ source file named `salario_funcionario.cpp`. The code defines a `main` function that calculates a salary adjustment. A red arrow points to a syntax error in the `scanf` function call on line 11: `scanf("%f", sal_func);`. The error is an ampersand (&) that should be removed, as `sal_func` is a float variable, not a pointer. The IDE's status bar at the bottom shows the compilation process, including the compiler name (TDM-GCC 4.9.2 64-bit Release) and the command used to compile the file.

```

salario_funcionario.cpp
2  #include <stdio.h>
3  #include <stdlib.h>
4  //Obrigatório. Função principal: indica o início da execução do programa
5  int main
6  {
7      //Declaração de Variáveis Locais.
8      float sal_func, sal_reajuste;
9      //Obrigatório. Comandos para resolução do problema
10     printf("Digite o salário do funcionário: ");
11     scanf("%f", sal_func);
12     if (sal_func < 500)
13     {
14         sal_reajuste = sal_func + sal_func * 0.15;
15     }
16 }

```

Compiling single file...

-----

- Filename: G:\BKP\Users\User\Desktop\Backup\25\_02\_2011\Leandro\BACKUP\Aulas\l

- Compiler Name: TDM-GCC 4.9.2 64-bit Release

Processing C++ source file...

-----

- C++ Compiler: C:\Program Files (x86)\Dev-Cpp\MinGW64\bin\g++.exe

- Command: g++.exe "G:\BKP\Users\User\Desktop\Backup\25\_02\_2011\Leandro\BACKUP"

Compilation results...

-----

# Estratégia para definir um algoritmo







- Codificação: Alguns Exemplos de Erros, usando Dev-C++ 5.11

salario\_funcionario.cpp

```

5  int main ()
6  {
7      //Declaração de Variáveis Locais.
8      float sal_func, sal_reajuste;
9      //Obrigatório. Comandos para resolução do problema
10     printf("Digite o salário do funcionário: ");
11     scanf("%f",&sal_func);
12     if (sal_func<500)
13     {
14         sal_reajuste=sal_func+sal_func*0.15;
15     }
16     else if (sal_func>=500 && sal_func<=1000)
17     {
18         sal_reajuste=sal_func+sal_func*0.10;
19     }
20     else
21     {
22         sal_reajuste=sal_func+sal_func*0.05;
23     }
24     printf("\n0 salário do funcionário é: %.2f", sal_func);
25     printf("\n0 salário com reajuste é: %.2f", sal_reajuste);
26     //Opcional. Comando para interromper momentaneamente o programa

```

 Compilador (2)
  Recursos
  Registro do Compilador
  Depurador
  Resultados da Busca
  Fechar

Lin...	Col...	Unidade	Mensagem
		G:\BKP\Users\User\Desktop\Backup\25_02_2011\Le...	In function 'int main()':
16	2	G:\BKP\Users\User\Desktop\Backup\25_02_2011\Leandr...	[Error] expected '}' before 'else'

# Estratégia para definir um algoritmo

## ■ Codificação, Variações: Exemplo 1

salario\_funcionario\_ajustada.cpp

```
2  #include <stdio.h>
3  #include <stdlib.h>
4  //Obrigatório. Função principal: indica o início da execução do programa
5  int main ()
6  {
7      //Declaração de Variáveis Locais.
8      float sal_func, sal_reajuste;
9      //Obrigatório. Comandos para resolução do problema
10     printf("Digite o salário do funcionário: ");
11     scanf("%f",&sal_func);
12     sal_reajuste=sal_func;
13     if (sal_func<500)
14         sal_reajuste+=sal_func*0.15;
15     else if (sal_func>=500 && sal_func<=1000)
16         sal_reajuste+=sal_func*0.10;
17     else
18         sal_reajuste+=sal_func*0.05;
19
20     printf("\nO salário do funcionário é: %.2f", sal_func);
21     printf("\nO salário com reajuste é: %.2f", sal_reajuste);
22     //Opcional. Comando para interromper momentaneamente o programa
23     printf("\n");
24     system("PAUSE");
25     //Retorno ao SO o status do programa
26     return 0;
27 } //Indica o final do programa.
```

# Estratégia para definir um algoritmo

## ■ Codificação, Variações: Exemplo 2

salario\_funcionario\_ajustado.cpp

```
1 //Diretivas de Pré-processamento (Obrigatórias)
2 #include <stdio.h>
3 #include <stdlib.h>
4 //Obrigatório. Função principal: indica o início da execução do programa
5 int main ()
6 {
7     //Declaração de Variáveis Locais.
8     float sal_func, sal_reajuste;
9     //Obrigatório. Comandos para resolução do problema
10    printf("Digite o salário do funcionário: ");
11    scanf("%f",&sal_func);
12
13    if (sal_func<500)
14        sal_reajuste=sal_func*0.15;
15    else if (sal_func>=500 && sal_func<=1000)
16        sal_reajuste=sal_func*0.10;
17    else
18        sal_reajuste=sal_func*0.05;
19
20    printf("\nO salário do funcionário é: %.2f", sal_func);
21    printf("\nO salário com reajuste é: %.2f", sal_func+sal_reajuste);
22    //Opcional. Comando para interromper momentaneamente o programa
23    printf("\n");
24    system("PAUSE");
25    //Retorno ao SO o status do programa
26    return 0;
27 } //Indica o final do programa.
```

# Estratégia para definir um algoritmo

## Codificação, Variações: Exemplo 3

Atenção:  
instruções  
redundantes

salario\_funcionario\_ajustada\_3.cpp

```

1 //Diretivas de Pré-processamento (Obrigatórias)
2 #include <stdio.h>
3 #include <stdlib.h>
4 //Obrigatório. Função principal: indica o início da execução do programa
5 int main ()
6 {
7     //Declaração de Variáveis Locais.
8     float sal_func;
9     //Obrigatório. Comandos para resolução do problema
10    printf("Digite o salário do funcionário: ");
11    scanf("%f",&sal_func);
12
13    if (sal_func<500)
14    {
15        printf("\nO salário do funcionário é: %.2f", sal_func);
16        printf("\nO salário com reajuste é: %.2f", sal_func+sal_func*0.15);
17    }
18    else if (sal_func>=500 && sal_func<=1000)
19    {
20        printf("\nO salário do funcionário é: %.2f", sal_func);
21        printf("\nO salário com reajuste é: %.2f", sal_func+sal_func*0.10);
22    }
23    else
24    {
25        printf("\nO salário do funcionário é: %.2f", sal_func);
26        printf("\nO salário com reajuste é: %.2f", sal_func+sal_func*0.05);
27    }
28
29    //Opcional. Comando para interromper momentaneamente o programa
30    printf("\n");
31    system("PAUSE");
32    //Retorno ao SO o status do programa
33    return 0;
34 } //Indica o final do programa.
  
```

# Estratégia para definir um algoritmo

## Codificação, Variações: Exemplo 4

salario\_funcionario\_ajustado\_4.cpp

```

1 //Diretivas de Pré-processamento (Obrigatórias)
2 #include <stdio.h>
3 #include <stdlib.h>
4 //Obrigatório. Função principal: indica o início da execução do programa
5 int main ()
6 {
7     //Declaração de Variáveis Locais.
8     float sal_func, sal_reajuste;
9     //Obrigatório. Comandos para resolução do problema
10    printf("Digite o salário do funcionário: ");
11    scanf("%f",&sal_func);
12
13    if (sal_func<500)
14        sal_reajuste=sal_func*1.15;
15    else if (sal_func>=500 && sal_func<=1000)
16        sal_reajuste=sal_func*1.10;
17    else
18        sal_reajuste=sal_func*1.05;
19
20    printf("\nO salário do funcionário é: %.2f", sal_func);|
21    printf("\nO salário com reajuste é: %.2f", sal_reajuste);
22    //Opcional. Comando para interromper momentaneamente o programa
23    printf("\n");
24    system("PAUSE");
25    //Retorno ao SO o status do programa
26    return 0;
27 }//Indica o final do programa.
  
```

# Exemplo de compilação no terminal

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Olá, usando o gcc no linux!");
5     return 0;
6 }
```

```
❖ clang-7 -pthread -lm -o main main.c
❖ ./main
Olá, usando o gcc no linux!❖
```

- gcc <nome do arquivo> -o <nome executável – arquivo binário>
- ./<nome executável – arquivo binário>

```
❖ clang-7 -pthread -lm -o main main.c
❖ ./main
Hello, world!❖
❖ dir
main main.c
❖ ./main
Hello, world!❖
❖ rm main
❖ dir
main.c
❖ mv main.c teste.c
❖ dir
teste.c
❖ gcc teste.c -o teste
❖ ls
teste teste.c
❖ gcc teste.c -o prog
❖ ls
prog teste teste.c
❖ ./prog
Hello, world!❖
```



# Exemplo de compilação no terminal

```

1  #include <stdio.h>
2  #include <math.h>
3
4  int main() {
5      double a=2;
6      printf("%lf", pow(a,a));
7
8      return 0;
9  }
10

```

```

❯ clang-7 -pthread -lm -o main main.c
❯ ./main
4.000000❯
❯ ls
main  main.c
❯ gcc main.c -o main
/tmp/ccxvgFmF.o: In function `main':
main.c:(.text+0x2c): undefined reference to `pow'
collect2: error: ld returned 1 exit status
❯

```

- gcc <nome do arquivo> **-lm** -o <nome executável – arquivo binário>

**-lm**: inclusão de bibliotecas na linkedição

```

❯ clang-7 -pthread -lm -o main main.c
❯ ./main
4.000000❯
❯ ls
main  main.c
❯ gcc main.c -o main
/tmp/ccxvgFmF.o: In function `main':
main.c:(.text+0x2c): undefined reference to `pow'
collect2: error: ld returned 1 exit status
❯
❯ gcc main.c -lm -o main
❯ ls
main  main.c
❯

```

- Até aqui vimos o seguinte:
  - Estratégias para Codificação
  - Próximo Conteúdo:
    - Estrutura de Repetição

# Bibliografia Complementar

- SCHILDT, H. C Completo e Total, 3ª ed., Pearson 1996. 852p.
  - Páginas 61 a 74, até o tópico switch aninhados
  
- SALES, André Barros de; AMVAME-NZE, Georges Daniel. Linguagem C: roteiro de experimentos para aulas práticas [recurso eletrônico]. Florianópolis: UFSC, 2016. Disponível em: <<http://repositorio.unb.br/handle/10482/21540>>.
  - Páginas 35 a 53
  - Realizar os Experimentos e Atividades de Fixação

