

Técnicas de Desenvolvimento de Algoritmos



Cruzeiro do Sul Virtual
Educação a distância

Material Teórico



Sub-rotinas

Responsável pelo Conteúdo:

Prof.^a Me. Ana Fernanda Gomes Ascencio

Revisão Textual:

Jaquelina Kutsunugi

Revisão Técnica:

Prof.^a Esp. Margarete Eliane da Silva Almendro



- Sub-rotina



OBJETIVO DE APRENDIZADO

- Desenvolver algoritmos em fluxograma e em pseudocódigo com sub-rotinas com e sem parâmetros e com e sem retorno.

Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:



Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como seu “momento do estudo”;
- ✓ Procure se alimentar e se hidratar quando for estudar; lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo;
- ✓ No material de cada Unidade, há leituras indicadas e, entre elas, artigos científicos, livros, vídeos e sites para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados;
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e de aprendizagem.

Sub-rotina

Nesta unidade V, continuaremos a conhecer um pouco mais sobre as estruturas que compõem os algoritmos. Salientamos que a implementação dos algoritmos utilizando qualquer Linguagem de Programação permite o teste efetivo destes. Neste curso, os testes serão realizados na ferramenta VisualG.

Na Unidade I, abordamos as definições de algoritmos e as três técnicas mais utilizadas para o desenvolvimento dos mesmos. Na Unidade II, foram abordados alguns algoritmos que utilizam apenas estrutura sequencial, ou seja, entrada dos dados, processamento e saída. Na Unidade III, abordamos alguns algoritmos que utilizam estruturas condicionais, também conhecidas como desvios. Na unidade IV, por fim, abordamos alguns algoritmos que utilizam estruturas de repetição, em que um ou vários comandos podem ser repetidos várias vezes.

A partir desta unidade, abordaremos a modularização dos algoritmos, ou seja, entenderemos como as instruções que compõem os algoritmos podem ser divididas em blocos menores e com objetivos bem definidos.

A nomenclatura adotada para a modularização depende do paradigma de programação que está sendo adotado, assim, na programação estruturada, a modularização é composta por procedimentos e funções também chamados de sub-rotinas, e na programação orientada a objetos, a modularização é conhecida como métodos.

As sub-rotinas, também chamadas de subprogramas, são blocos de instruções que realizam tarefas específicas. O código de uma sub-rotina é carregado uma vez e pode ser executado quantas vezes forem necessárias. Como o problema pode ser subdividido em pequenas tarefas, os programas tendem a ficar menores e mais organizados. Os programas em geral são executados linearmente, uma linha após a outra, até o fim. Entretanto, quando são utilizadas sub-rotinas, é possível a realização de desvios na execução dos programas. Estes desvios são efetuados quando uma função é chamada pelo programa principal. (ASCENCIO; CAMPOS, 2012, p. 252)

Em uma linguagem que dá suporte ao paradigma orientado a objeto, todos os programas devem fazer uso de classes. Uma classe pode ser entendida como um tipo de dado capaz de armazenar diversas informações e também várias funções para manipular adequadamente essas informações. Seguindo o paradigma orientado a objetos, essas informações são chamadas de atributos, e as funções, de métodos. (ASCENCIO; CAMPOS, 2012, p. 271)

Neste material, adotaremos a nomenclatura sub-rotinas e, independente dessa nomenclatura, as sub-rotinas aqui abordadas se dividem em quatro tipos:

- a) sub-rotinas sem parâmetros;
- b) sub-rotinas com parâmetros;

- c) sub-rotinas sem retorno;
- d) sub-rotinas com retorno.

Antes de nos aprofundar nos tipos de sub-rotinas, é necessário compreender duas definições muito importantes: variáveis globais e variáveis locais ou internas.

As **VARIÁVEIS GLOBAIS** são declaradas logo após as palavras algoritmo “nome” e são acessíveis por qualquer parte do algoritmo, ou seja, são acessíveis pelas instruções do algoritmo principal e pelas instruções das sub-rotinas.

As **VARIÁVEIS LOCAIS** ou **INTERNAS** são declaradas dentro de cada sub-rotina e são acessíveis apenas pelas instruções das sub-rotinas.



O uso de variáveis globais deve ser feito com moderação, pois isso dificulta a manutenção e a busca por erros nos algoritmos, tendo em vista que as variáveis globais podem ser alteradas por qualquer parte do código.

O primeiro tipo de sub-rotina a ser explorada trata-se das sub-rotinas sem parâmetros e sem retorno, também conhecidas como procedimentos, são consideradas apenas um desvio para a execução de um conjunto de instruções bem definidas.

As sub-rotinas são representadas em fluxograma e em pseudocódigos, conforme Quadro 1.

Quadro 1 - Sub-rotinas em fluxograma e em pseudocódigo

Fluxograma	Pseudocódigo
<div style="border: 2px solid red; padding: 10px; text-align: center;"> Nome da Sub-rotina </div>	<pre> algoritmo "nome" // declaração de variáveis globais procedimento "nome_da_sub_rotina" // declaração de variáveis locais início //comandos_da_sub_rotina fimprocedimento início comandos_do_corpo_principal_do_algoritmo // desvio para execução da sub_rotina nome_da_sub_rotina fimalgoritmo </pre>

Exemplo 1 - Faça um algoritmo para calcular as quatro operações básicas entre dois números, em que cada operação deve ser realizada em um sub-rotina apropriada.

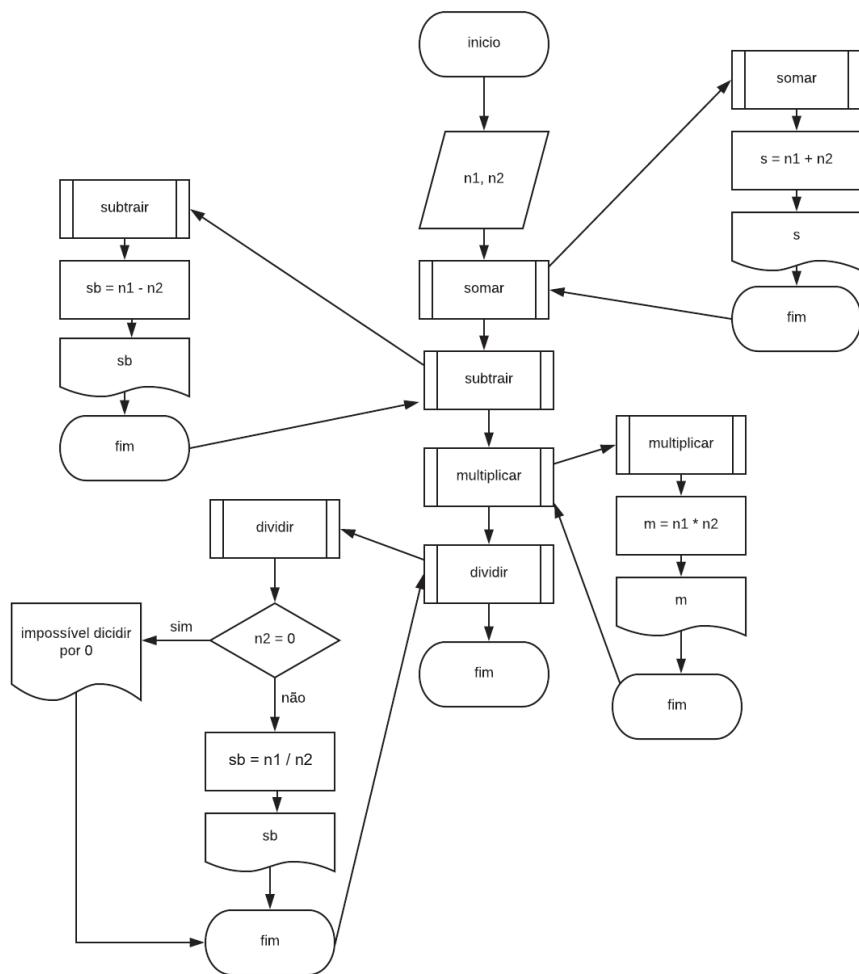


Figura 1 - Fluxograma do algoritmo do Exemplo 1

A seguir, o Exemplo 1 em pseudocódigo.

Algoritmo "Exemplo1"

```

// Função: calcular as quatro operações entre dois números
// Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
// Data: 24/12/2018      /*07/02/2020
// seção de declarações de variáveis globais

var n1, n2: real

// a seguir a sub-rotina para somar dois números
procedimento somar

```

```
// declaração de variáveis locais do procedimento somar
var s: real
inicio
  s <- n1 + n2
  escreval ("soma = ", s)
fimprocedimento

// a seguir a sub-rotina para subtrair dois números
procedimento subtrair
// declaração de variáveis locais do procedimento sub-
trair
var sb: real
inicio
  sb <- n1 - n2
  escreval ("subtração = ", sb)
fimprocedimento

// a seguir a sub-rotina para multiplicar dois números
procedimento multiplicar
// declaração de variáveis locais do procedimento mul-
tiplicar
var m: real
inicio
  m <- n1 * n2
  escreval ("multiplicação = ", m)
fimprocedimento

// a seguir a sub-rotina para dividir dois números
procedimento dividir
```

```
// declaração de variáveis locais do procedimento dividir  
  
var d: real  
  
inicio  
  
se (n2 = 0) entao  
  
    escreval("impossível dividir por zero")  
  
senao inicio  
  
    d <- n1 / n2  
  
    escreval("divisão = ",d)  
  
fimse  
  
fimprocedimento  
  
// início do programa principal  
  
inicio  
  
    escreval ("digite o primeiro número ")  
  
    leia (n1)  
  
    escreval ("digite o segundo número ")  
  
    leia (n2)  
  
    somar  
  
    subtrair  
  
    multiplicar  
  
    dividir  
  
fimalgoritmo
```



O uso de sub-rotina torna os códigos mais organizados e consequentemente de mais fácil entendimento e manutenção. Sempre que possível faça o uso de sub-rotinas, pois nem sempre quem desenvolveu o algoritmo será quem dará a sua manutenção.

O segundo tipo de sub-rotina a ser explorada trata-se das sub-rotinas com parâmetros e sem retorno, também conhecidas como procedimentos, são consideradas um desvio para a execução de um conjunto de instruções bem definidas e, neste desvio, são fornecidos valores para a subrotina.

Exemplo 2 - Faça um algoritmo para calcular o valor do imposto e o valor do desconto de um salário. Sabe-se que o imposto de é 7% e o desconto é de 5%, em que cada operação deve ser realizada em um sub-rotina apropriada.

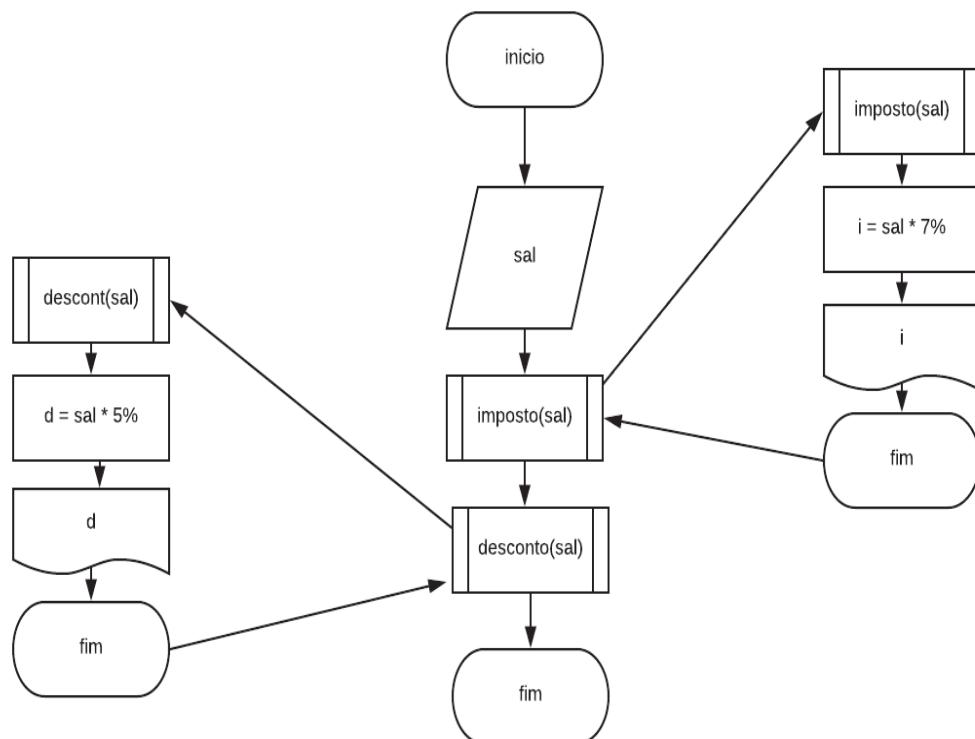


Figura 2 - Fluxograma do algoritmo do Exemplo 2



Não é possível ter mais de uma sub-rotina com o mesmo nome, mas é possível chamar uma sub-rotina para execução quantas vezes forem necessárias.

A seguir, o Exemplo 2 em pseudocódigo.

algoritmo “exemplo2”

```

// Função: calcular desconto e imposto
// Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
// Data: 24/12/2018      /*07/02/2020
// seção de declarações de variáveis globais
var sal: real
  
```

```
// a seguir a sub-rotina para calcular o imposto
procedimento imposto (s:real)
    // declaração de variáveis locais do procedimento
    imposto
        var i: real
        inicio
            i <- s * 7/100
            escreval ("imposto = ",i)
        fimprocedimento

// a seguir a sub-rotina para calcular o desconto
procedimento desconto (s:real)
    // declaração de variáveis locais do procedimento
    desconto
        var d: real
        inicio
            d <- s * 5/100
            escreval ("desconto = ",d)
        fimprocedimento

// início do programa principal
inicio
    escreval ("digite o salário ")
    leia (sal)
    imposto (sal)
    desconto (sal)
fimalgoritmo
```

O terceiro tipo de sub-rotina a ser explorada trata-se das sub-rotinas sem parâmetros e com retorno, também conhecidas como funções, são consideradas um

desvio para a execução de um conjunto de instruções bem definidas e como as funções SEMPRE retornam um valor, devem ser atribuídas a uma variável.

Exemplo 3 - Faça um algoritmo para calcular a área de um quadrado. O cálculo deve ser feito em uma função.

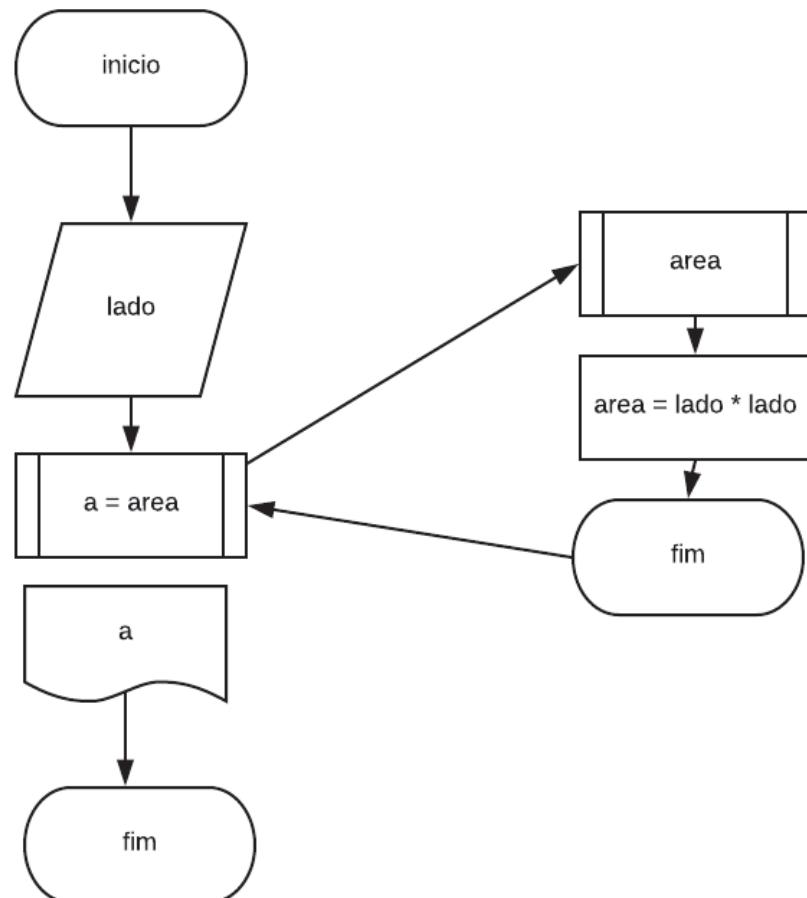


Figura 3 - Fluxograma do algoritmo do Exemplo 3

A seguir, o Exemplo 3 em pseudocódigo.

```

algoritmo "exemplo3"
    // Função: calcular área do quadrado
    // Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
    // Data: 24/12/2018      /*07/02/2020
    // seção de declarações de variáveis globais
    var lado, a: real
        // a seguir a sub-rotina para calcular a área
        // como tem retorno será uma função
    
```

```
funcao area(Retorno: real):real
var resp:real
inicio
    resp <- lado * lado
    retorno resp
fimfunção

// início do programa principal
inicio
    escreval ("digite o lado ")
    leia (lado)
    a <- area(lado)
    escreval ("área = ", a)
fimalgoritmo
```



As linguagens de programação tem várias sub-rotinas já implementadas e que podem ser utilizadas nos programas desenvolvidos pelo usuário. Um exemplo dessas sub-rotinas são as que manipulam cadeias de caracteres.

O quarto tipo de sub-rotina a ser explorada trata-se das sub-rotinas com parâmetros e com retorno, também conhecidas como funções, são consideradas um desvio para a execução de um conjunto de instruções bem definidas e como as funções SEMPRE retornam um valor, devem ser atribuídas à uma variável. Neste caso, além do desvio são fornecidos valores para a subrotina.

Exemplo 4 - Faça um algoritmo para calcular o novo salário sabendo que este tem aumento de 10% e imposto é de 6% já sobre o salário reajustado. Os cálculos devem ser feitos em funções.

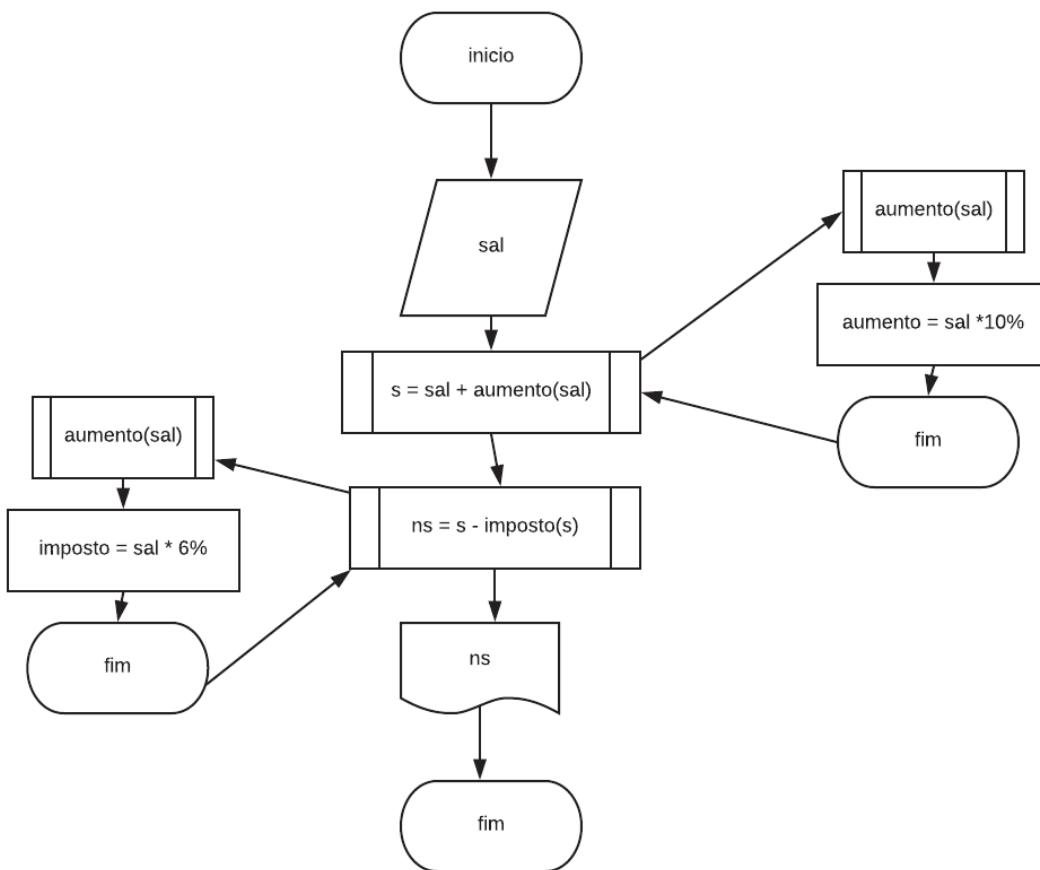


Figura 4 - Fluxograma do algoritmo do Exemplo 4

A seguir, o Exemplo 4 em pseudocódigo.

```

algoritmo "exemplo4"
  // Função: calcular o novo salário
  // Autor: Ana Fernanda /*revisão Margarete E. S. Almendro
  // Data: 24/12/2018      /*07/02/2020
  // seção de declarações de variáveis globais
  var s, sal, ns: real
    // a seguir a sub-rotina para calcular o aumento
    // como tem retorno será uma função
    funcao aumento (sal: real) :real
  
```

```

var aum:real
inicio
    aum <- sal * 0.1
    retorno aum
fimfunção

função imposto (sal: real) :real
var imp:real
inicio
    imp <- sal * 0.06
    retorno imp
fimfunção

// início do programa principal
inicio
    escreval ("digite o salário")
    leia (sal)
    s <- sal + aumento(sal)
    escreval ("Aumento = ",s)
    ns <- s - imposto(s)
    escreval ("Novo salário = ",ns)
fimalgoritmo

```

O infográfico, a seguir, ilustra os quatro tipos de sub-rotinas, tipos estes abordados nessa unidade.

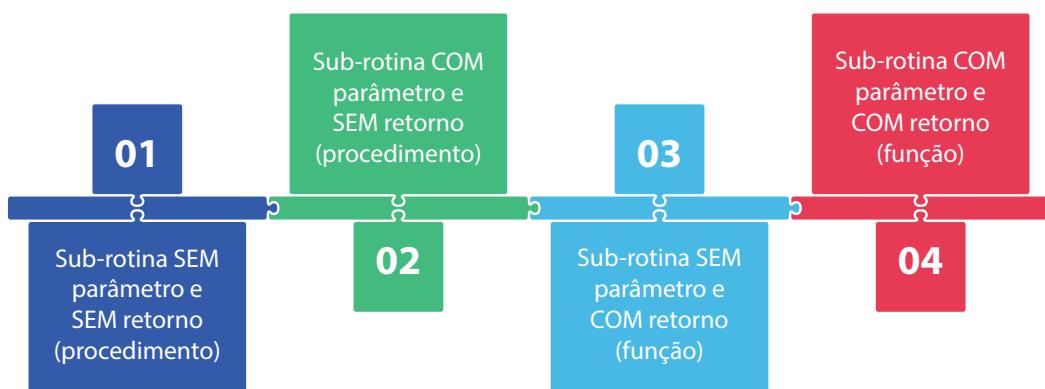


Figura 5 - Tipos de Sub-rotinas

Nesta unidade, exploramos as sub-rotinas e pudemos perceber como os algoritmos modularizados facilitam a manutenção e a correção de erros.

Material Complementar

Indicações para saber mais sobre os assuntos abordados nesta Unidade:



Livros

Lógica de Programação com Pascal

O livro **Lógica de Programação com Pascal**, no Capítulo 8, mostra a teoria sobre sub-rotinas e vários exemplos. Ao final desse capítulo, há uma lista de exercícios a resolver. **ASCENCIO, A. F. G.; CAMPOS, E. A. V. Fundamentos da Programação de Computadores.** 3 ed. São Paulo: Pearson, 2012.

Fundamentos da Programação de Computadores

O livro **Fundamentos da Programação de Computadores**, no Capítulo 8, mostra a teoria sobre sub-rotinas e vários exemplos. Ao final desse capítulo, existem duas listas de exercícios, uma resolvida e a outra a resolver.

ASCENCIO, A. F. G.; CAMPOS, E. A. V. Fundamentos da Programação de Computadores. 3 ed. São Paulo: Pearson, 2012.

Algoritmos: Teoria e Prática

O livro **Algoritmos, Teoria e Prática**, do autor Thomas Cormen, é um clássico na área de Computação e aborda o desenvolvimento de algoritmos na teoria e na prática. **CORMEN, T. H. Algoritmos, Teoria e Prática.** Rio de Janeiro: Campus, 2012.

Algoritmos

O livro **Algoritmos**, de José Augusto Manzano, é muito indicado para quem está começando a desenvolver algoritmos, tendo em vista que possível sequência didática de aprendizagem e uma linguagem apropriada para iniciantes.

MANZANO, J. A. N. G. Algoritmos. 28 ed. São Paulo: Pearson, 2016.

Referência

ASCENCIO, A. F. G.; CAMPOS, E. A. V. **Fundamentos da Programação de Computadores**. 3 ed. São Paulo: Pearson, 2012.



Cruzeiro do Sul
Educacional