Técnica de Desenvolvimento de Algoritmos

Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

Material Teórico

Organização Básica de um computador

Responsável pelo Conteúdo:
Prof. Esp. Alexander Gobbato Albuquerque

Revisão Textual:
Profa. Esp. Vera Lídia de Sá Cicaroni

UNIDADE Organização Básica de um computador

- Organização Básica de um computador

Objetivo de APRENDIZADO

- Nesta unidade, estudaremos os conceitos básicos para a criação de algoritmos, entenderemos o conceito de lógica aplicada a programas e utilizaremos pensamento crítico, operacional e lógico, através de modelos de representação de algoritmos

Hoje veremos alguns assuntos introdutórios à nossa disciplina e aproveito para apresentar-lhes alguns conceitos que utilizaremos na estrutura de todas as nossas unidades.

Para obter um bom aproveitamento nesta unidade, vamos conferir a estrutura desta unidade:

Conteúdo Teórico: neste link, você encontrará o material principal de estudos na forma de texto escrito.

Atividade de Sistematização: os exercícios disponibilizados são de autocorreção e visam a que você pratique o que aprendeu na disciplina e que identifique os pontos em que precisa prestar mais atenção ou sobre os quais necessita pedir esclarecimentos a seu tutor. Além disso, a esses exercícios serão atribuídas notas que farão parte de sua média final na disciplina.

Atividade de Aprofundamento: é uma atividade dissertativa ou de pesquisa.

Material Complementar e Referências Bibliográficas: nestes links, você poderá ampliar seus conhecimentos.

Vídeoaula: aqui serão apresentadas algumas ferramentas na prática e também a resolução de alguns exercícios de forma prática.

Lembramos a você a importância de realizar todas as atividades propostas dentro do prazo estabelecido para cada unidade. Dessa forma, você evitará que o conteúdo se acumule e que você tenha problemas ao final do semestre.

Contextualização

Aristóteles (384 a.C. – 322 a.C.) foi um filósofo grego responsável por escrever os primeiros grandes trabalhos de lógica, Gottfried Wilhelm Leibniz (1646–1716) propôs o uso e utilizou símbolos para processar o raciocínio dedutivo e George Boole (1815–1864) e Augustus de Morgan (1806–1871) propuseram as bases da lógica simbólica a partir dos símbolos de Leibniz.

Existem várias formas de representar uma lógica e é isto que iremos verificar nesta unidade: como funciona um computador, conceitos e formas de representar algoritmos.

Organização Básica de um computador

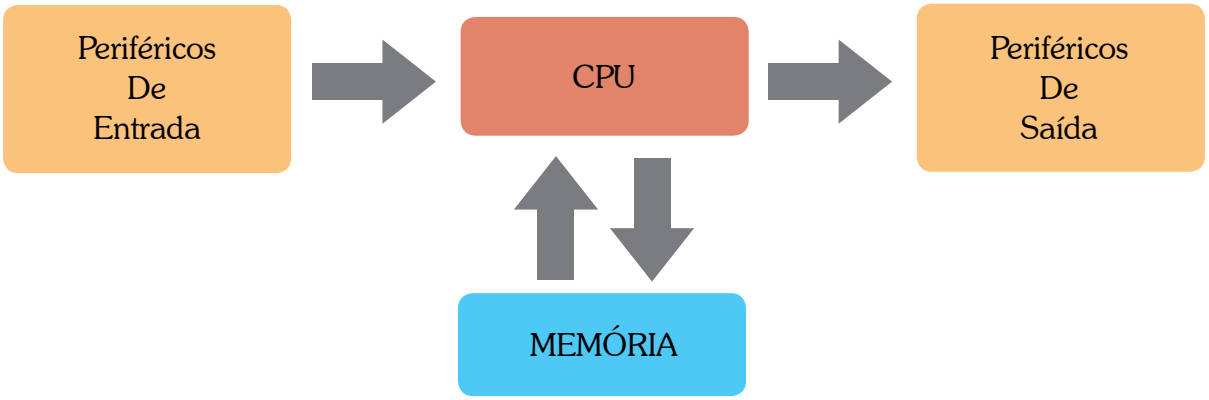
Um computador é constituído de quatro unidades básicas: unidade de entrada, unidade de saída, unidade de processamento central e memória. Como indica sua denominação, uma unidade de entrada é um dispositivo que permite que o usuário interaja com o computador, fornecendo-lhe dados e informações. O teclado e o mouse são os exemplos mais triviais de unidade de entrada. Uma unidade de saída, por sua vez, serve para que sejam fornecidos ao usuário do computador os resultados do processamento realizado.

O monitor de vídeo e uma impressora são exemplos de unidades de saída. A unidade central de processamento é responsável por todo o processamento requerido, sendo conhecida como CPU (central processing unit). Já a memória armazena dados e informações que serão utilizados no processamento, armazenamento temporário, pois, quando é desligado, toda a memória é apagada.

Linguagem de Máquina

Há a necessidade de que as unidades que compõem um computador se comuniquem umas com as outras. Por exemplo, um dado fornecido pelo teclado deve ser armazenado em memória para que a CPU o busque e realize as operações aritméticas.

Como a comunicação entre as unidades do computador teria que ser obtida através de fenômenos físicos, os cientistas que conceberam os computadores atuais estabeleceram dois símbolos básicos para a linguagem. Essa quantidade de símbolos foi escolhida pelo fato de que, através de fenômenos físicos, é muito fácil obter estados distintos e não confundíveis, como passar corrente elétrica e não passar, estar magnetizado ou não, etc. Assim, a linguagem utilizada para comunicação interna do computador (linguagem de máquina) possui apenas dois símbolos, sendo chamados de bit (binary digit) e representados por 0 ou 1.



Códigos

Para que haja a possibilidade de comunicação entre homem e máquina, é necessário que as palavras da linguagem sejam traduzidas para a linguagem de máquina e vice-versa. Para que isso seja possível, é necessário que se estabeleça qual sequência de bit´s corresponde a cada caractere usado na linguagem escrita. Um dos padrões de codificação bastante utilizado é o ASC II (American Standard Code for Information Interchange ou Código padrão Americano para Intercâmbio de Informação), estabelecido pelo ANSI (American National Standards Institute). Nessa codificação, cada caractere é representado por uma sequência de 8 bits (denominado byte). Cada caractere possui um código ASC diferente, incluindo a, A, á, ã, 1, #, &, -, }, etc. etc. Com 8 bits, temos 256 combinações diferentes de 0 a 255.

Podemos verificar esse fato acessando o código das teclas com o ALT esquerdo e uma sequência no teclado numérico. Como exemplo:

ALT + 65 = “A”

ALT + 66 = “B”

ALT + 33 = “!”

ALT + 166 = “ª”

ALT + 98 = “b”

E, assim, para cada caractere, temos um código, incluindo espaço, enter, ESC e etc.

Algoritmos

Consideraremos um algoritmo uma sequencia de instruções cuja execução resulta na realização de uma determinada tarefa. De uma maneira natural, alguns tipos de algoritmos estão presentes no nosso dia a dia, não necessariamente envolvendo aspectos computacionais. Uma receita de bolo, uma partitura musical, um manual de utilização do Playstation são algoritmos que descrevem, passo a passo, o que fazer para chegar a um resultado esperado.

Um algoritmo deve contemplar 3 exigências:

1) As instruções devem ser claras e não devem conter ambiguidades, nem qualquer coisa que impeça sua execução pelo processador.

2) Não pode haver dubiedade em relação à próxima ação a ser realizada após a execução de uma determinada ação.

3) Todas as instruções devem ser executadas num tempo finito.

Existem diversas formas de representação de algoritmos, mas não há um consenso com relação à melhor delas.

O critério usado para classificar hierarquicamente essas formas está diretamente ligado ao nível de detalhe ou, inversamente, ao grau de abstração oferecido.

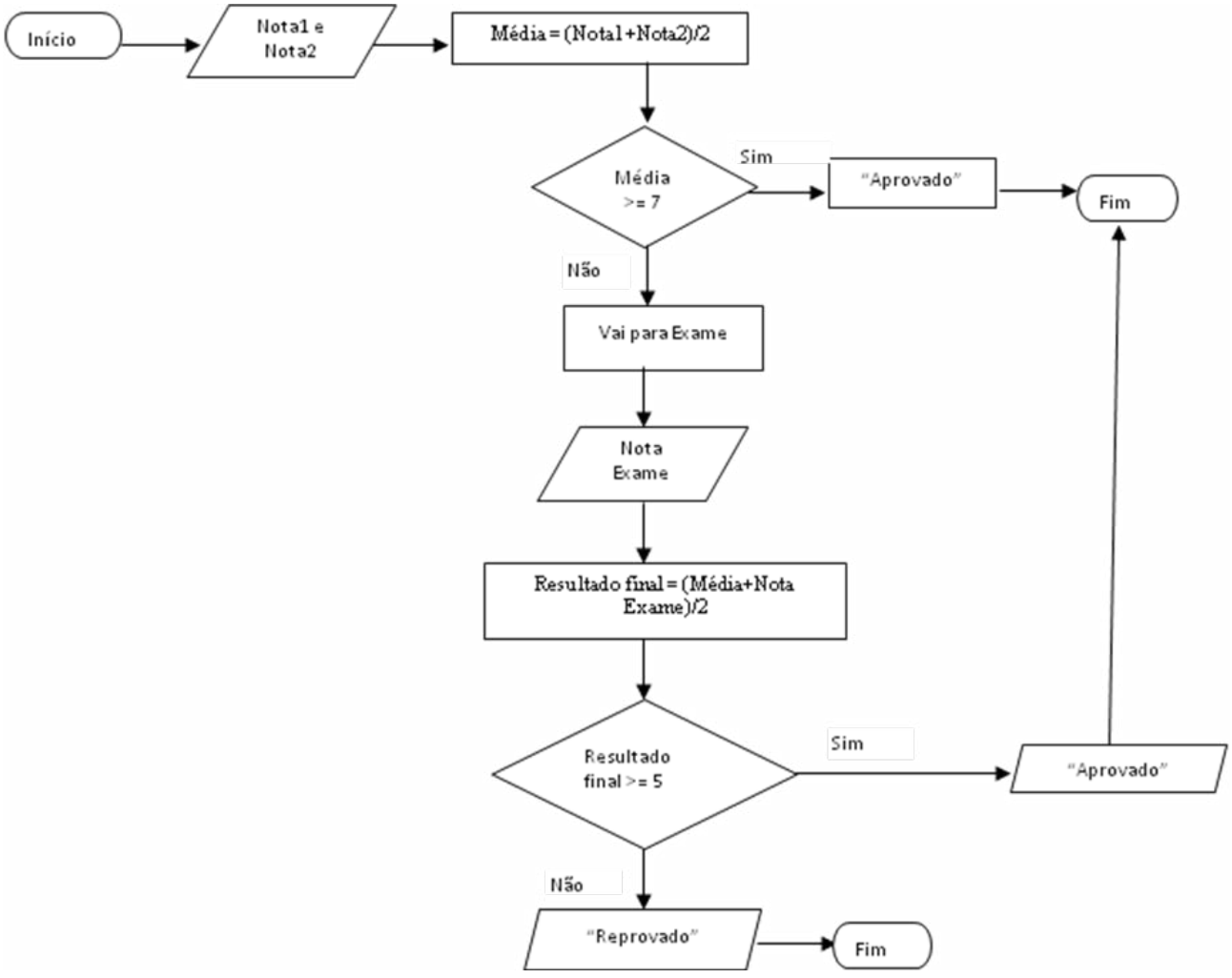
Algumas formas de representação de algoritmos tratam os problemas apenas em nível lógico, abstraindo-se de detalhes de implementação, muitas vezes, relacionados com alguma linguagem de programação específica. Por outro lado, existem formas de representação de algoritmos que possuem uma maior riqueza de detalhes e, muitas vezes, acabam por obscurecer as ideias principais do algoritmo, dificultando seu entendimento.

Dentre as formas de representação de algoritmos mais conhecidas podemos citar:

• Descrição Narrativa

- Ler 2 notas.
- Calcular a média entre elas.
- Imprimir aprovado se média for maior ou igual a sete.
- Se não resultar essa média, fazer exame.
- Se média entre Média final e exame for maior ou igual a 5, colocar aprovado; se não, colocar reprovado.

• Fluxograma Convencional



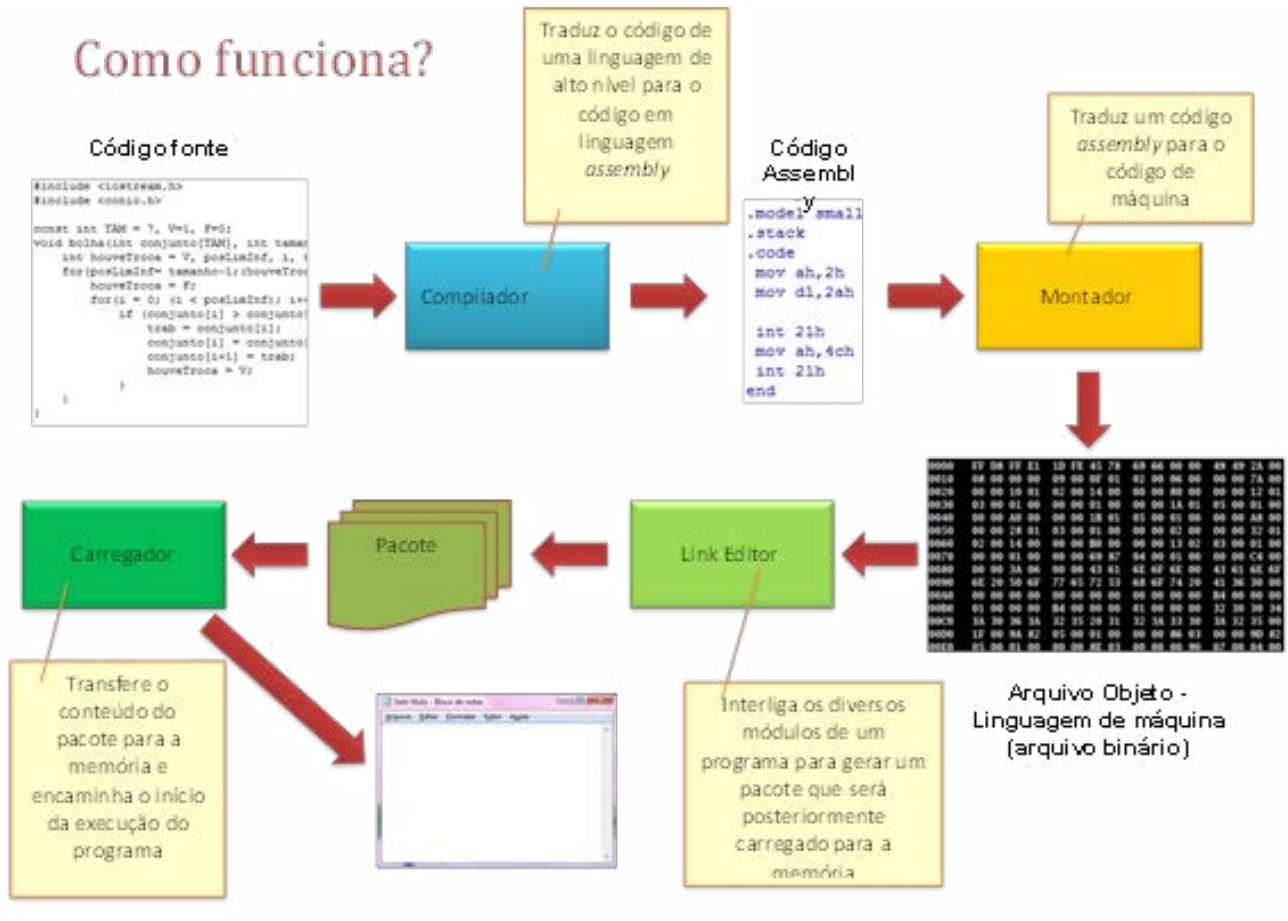
• Pseudocódigo, também conhecido como Português Estruturado ou Portugol.

```
início
      real media, nota1, nota2, exame, final;
      escreva "Digite a 1ª nota: ";
leia nota1;
escreva "Digite a 2ª nota: ";
leia nota2;
media ¬ (nota1 + nota2)/2;
      se (media >= 7)
            escreva "Aprovado";
      senão
            escreva "Digite a nota de exame";
            leia exame;
            final ¬ (media + exame)/2;
            se (final >= 5)
                  escreva "Aprovado";
            senão
                  escreva "Reprovado";
            fim se
      fim se
fim
```

• Usando uma linguagem de programação (Java)

```
import javax.swing.*;
public class Algoritmo {
  public static void main(String args[]) {
     float media, nota1, nota2, exame, final;
     nota1 = Float.parseFloat(JOptionPane.showInputDialog("Digite a 1ª nota:"));
     nota2 = Float.parseFloat(JOptionPane.showInputDialog("Digite a 2ª nota:"));
    media = (nota1 + nota2)/2;
        if (media >= 7)
           JOptionPane.showMessageDialog(null, "Aprovado");
    else {
      exame = Float.parseFloat(JOptionPane.showInputDialog(""Digite a nota de exame"));
       final = (media + exame)/2;
          if (final >= 5)
          JOptionPane.showMessageDialog(null, "Aprovado");
       else
          JOptionPane.showMessageDialog(null, "Reprovado");
    }
  }
}
```

Compilação

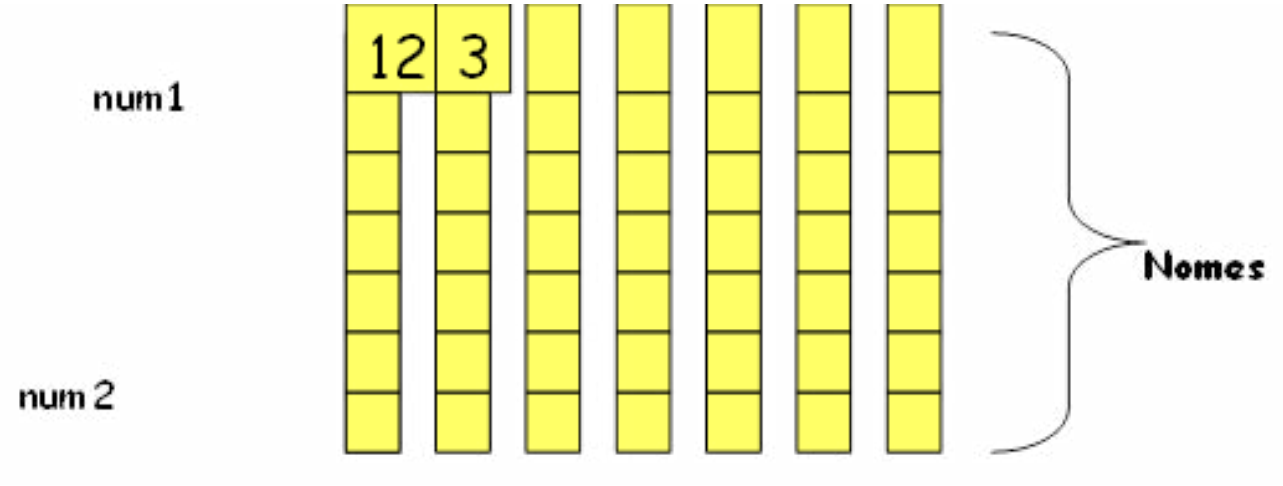


Variáveis

Anteriormente admitimos que o processador era capaz de associar cadeias de caracteres a valores numéricos (Código ASC II). Foi dito, também, que as informações devem ser armazenadas em memória para que o processador seja capaz de capturar as informações e fazer operações sobre eles.

Para que a memória possa armazenar dados, ela é dividida em partes, chamadas posições de memória. O sistema operacional que gerencia o sistema de computação pode acessar cada umas dessas posições de memória para armazenar os dados. Para que isso seja possível, cada uma delas está associada a um endereço em bytes.

Em programação, uma variável é uma posição de memória reservada pelo seu programa cujo conteúdo pode ser modificado durante a execução de um programa (em tempo de execução), devendo ser associado sempre a um identificador e um tipo de dado.



Identificador

O identificador é uma sequência de letras, dígitos e caractere underline (_) escolhidos pelo programador e será utilizado no programa para se fazer referência àquela variável. Como um programa deve ser legível por outros programadores, o nome do identificador deve ter relação com o conteúdo que será armazenado na variável. Por exemplo, para uma variável que armazenará a idade de uma pessoa, o identificador poderá se chamar "idade" e não "x".

Existem algumas regras para a criação de identificadores, ou seja, o identificador:

1) não poderá começar com números, que são permitidos apenas após letras ou o underline (_);

2) não poderá conter espaços em branco;

3) não poderá conter caracteres especiais como ç, á, õ, #, etc.;

4) não poderá ser uma palavra reservada da linguagem.

Exemplos de identificadores válidos;

minha_variavel;

sexo;

aluno22;

r2d2c3po;

Exemplos de identificadores inválidos:

meu nome; (espaço)
2r3poc; (começa com número)
opção; (caracteres especiais – acentos)
void; (palavra reservada do Java)

Tipos de Dados

Um tipo de dado associado a uma variável é um conjunto de elementos que podem ser armazenados nela. Ele diz qual o tipo de informação que será armazenada naquela área de memória. Utilizaremos 4 tipos de dados distintos:

1) inteiro: para armazenar valores inteiros, ou seja, que não possuem casas decimais. Nesse caso, não importa se o número é positivo ou negativo e nem há limites. O limite é apenas imposto quando escolhemos uma linguagem programação.

2) real: para armazenar valores reais, ou seja, que podem possuir ponto flutuante (casas decimais).

3) string: para armazenar cadeias de caracteres como palavras e/ou frases.

4) lógico: só consegue armazenar dois estados, sendo verdadeiro ou falso.

Exemplos de declaração de variáveis:

```
inteiro idade;
inteiro numero_filhos, quantidade_patas;
real salario;
real altura, peso;
string nome;
logico pagou;
```

Material Complementar

Explore

OLIVEIRA, J. F e MANZANO, J. A. N. G. **Algoritmos – Lógica para desenvolvimento de programação de computadores.** Ver o conteúdo de Parte II – Técnicas Básicas de Programação - Capítulo 3 – Tipo de Dados e Instruções Primitivas.

Referências

FARRER, H. **Algoritmos Estruturados.** 3. ed. Rio de Janeiro: Ltc-Livros Técnicos e Científicos, 1999.

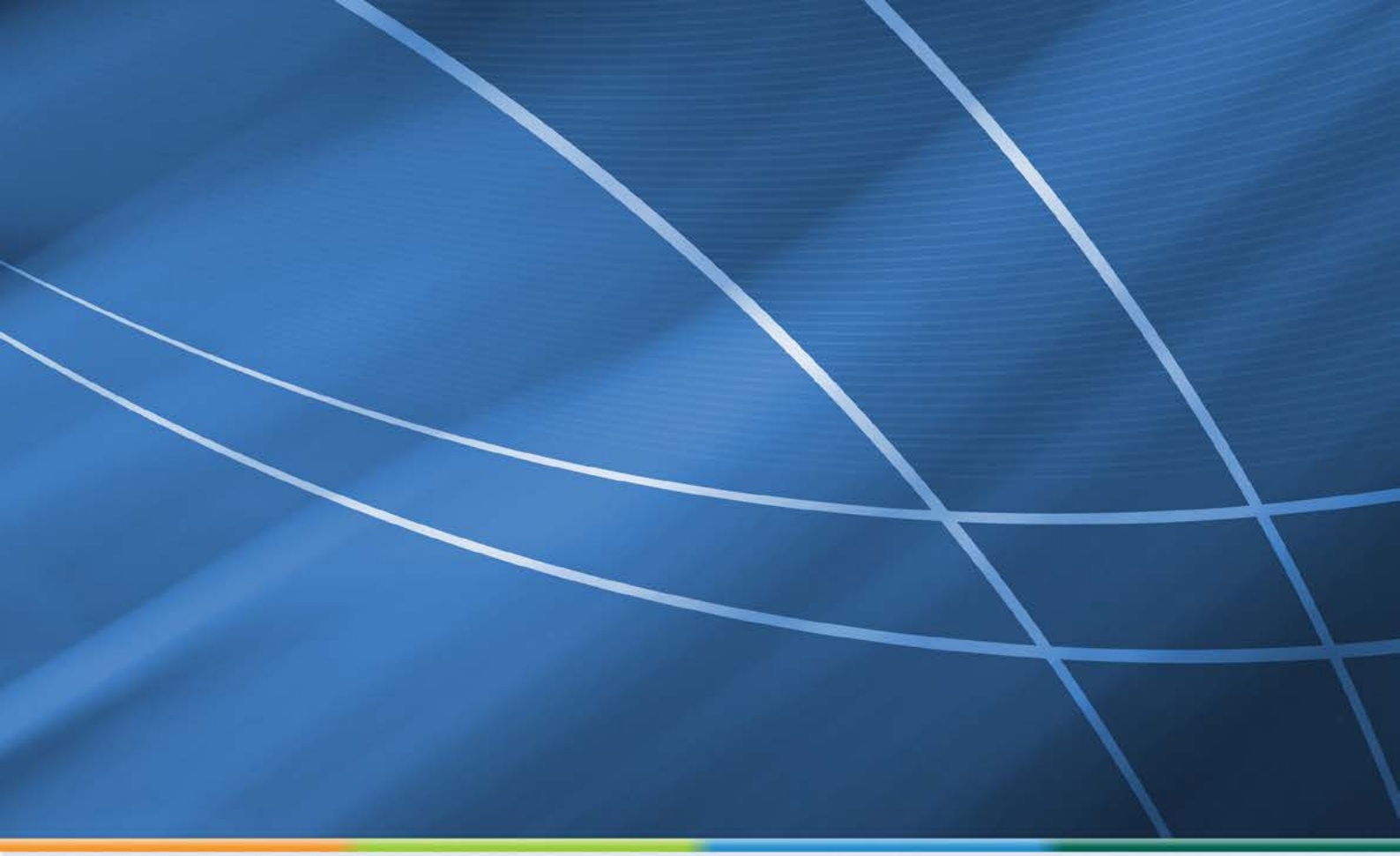
FORBELLONE, A. L. V.; EBERSPACHER, H. F. **Lógica de Programação: A Construção de Algoritmos e Estrutura de Dados.** 3. ed. São Paulo: Pearson Prentice Hall, 2008.

WIRTH, N. **Algoritmos e Estruturas de Dados.** Rio de Janeiro: Ltc-Livros Técnicos e Científicos, 1999.

Anotações

Educação a Distância
Cruzeiro do Sul Educacional
Campus Virtual

www.cruzeirodosulvirtual.com.br
Campus Liberdade
Rua Galvão Bueno, 868
CEP 01506-000
São Paulo SP Brasil
Tel: (55 11) 3385-3000



Universidade Cruzeiro do Sul

UNICID Universidade Cidade de S. Paulo

UNIFRAN Universidade de Franca

UDF Centro Universitário

Módulo Centro Universitário