Laboratório de Programação Avançada

2017.1

Ementa

- Linguagem de programação imperativa e bloco-estruturada: subprogramas, recursividade, arquivos, tipos de dados estruturados, alocação dinâmica de memória. Estruturas avançadas, pré-processador, modularização. Programação orientada a eventos. Estilo de programação. Atividades de desenvolvimento não abordadas em outras disciplinas que contemplem o caráter dinâmico e evolutivo da área de ciências da computação.

Conteúdo Programático

- Linguagem de programação imperativa e bloco-estruturada
  - Subprogramas;
  - Recursividade;
  - Arquivos;
  - Tipos de dados estruturados;
  - Alocação dinâmica de memória;
- Estruturas avançadas:
  - Listas
  - Filas
  - Pilhas
  - Arvores
  - Grafos
- Programação orientada a eventos
  - Gatilhos/Sensores
  - Interrupções
  - Eventos
- Programação Paralela
  - Arquitetura e Hardware
  - Threads
  - Comunicação entre processos e threads
  - Sincronismo e competição
  - Programação multi-core

Formas de avaliação

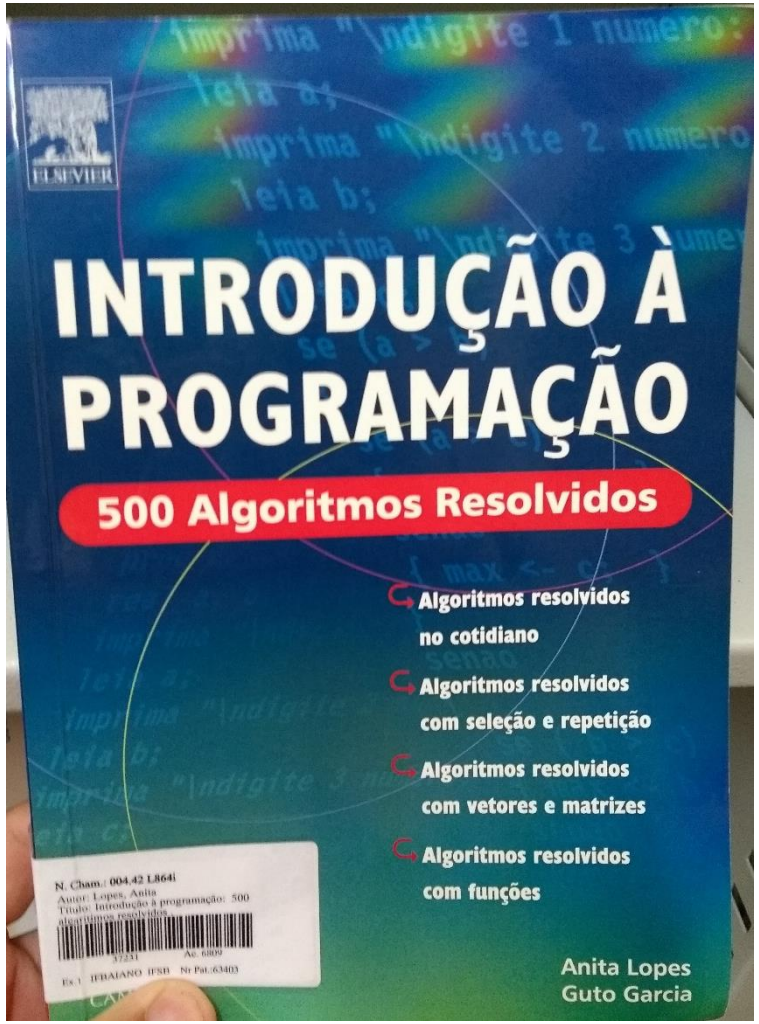
- Avaliação processual ao longo da disciplina através de participação em atividades práticas desenvolvidas em sala de aula;
- Avaliação de conteúdo prática por meio de Trabalho;
- Seminários de conteúdos teóricos-práticos.

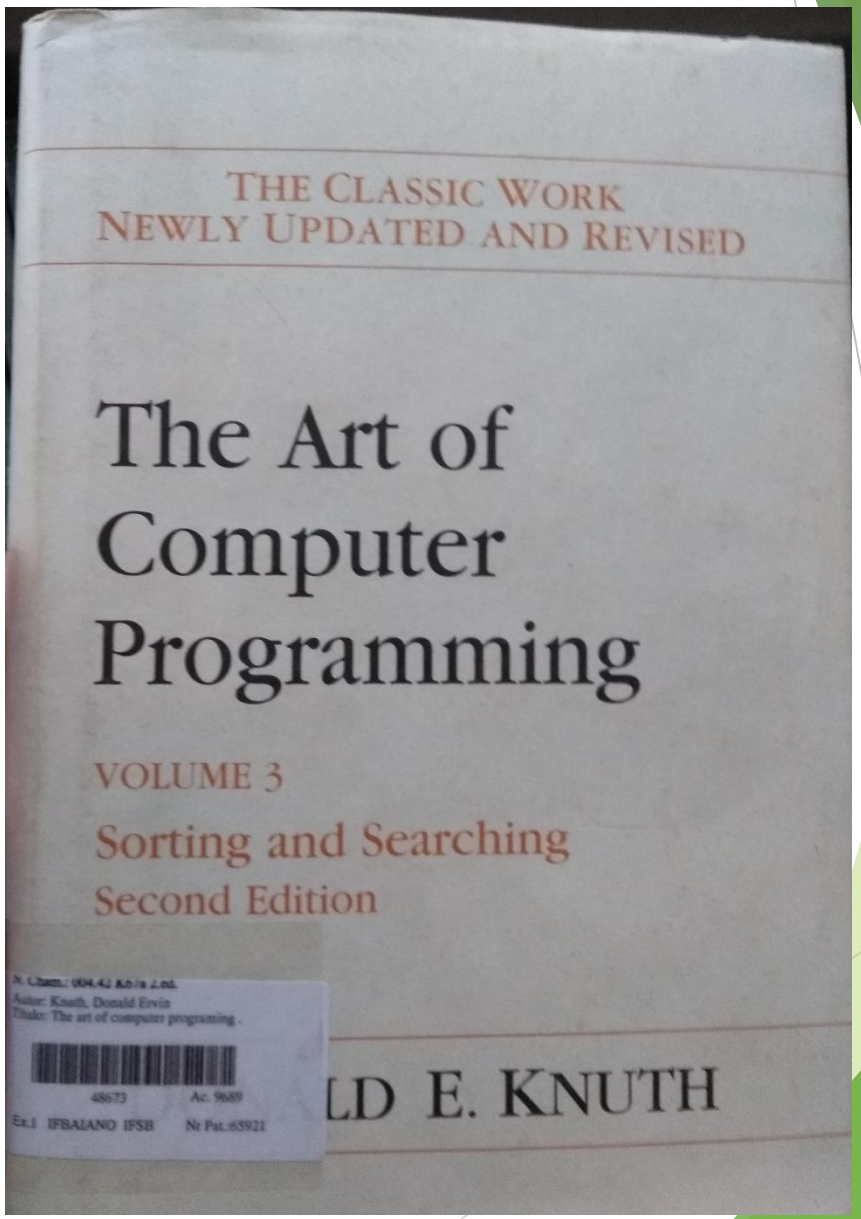
Bibliografia Recomendada

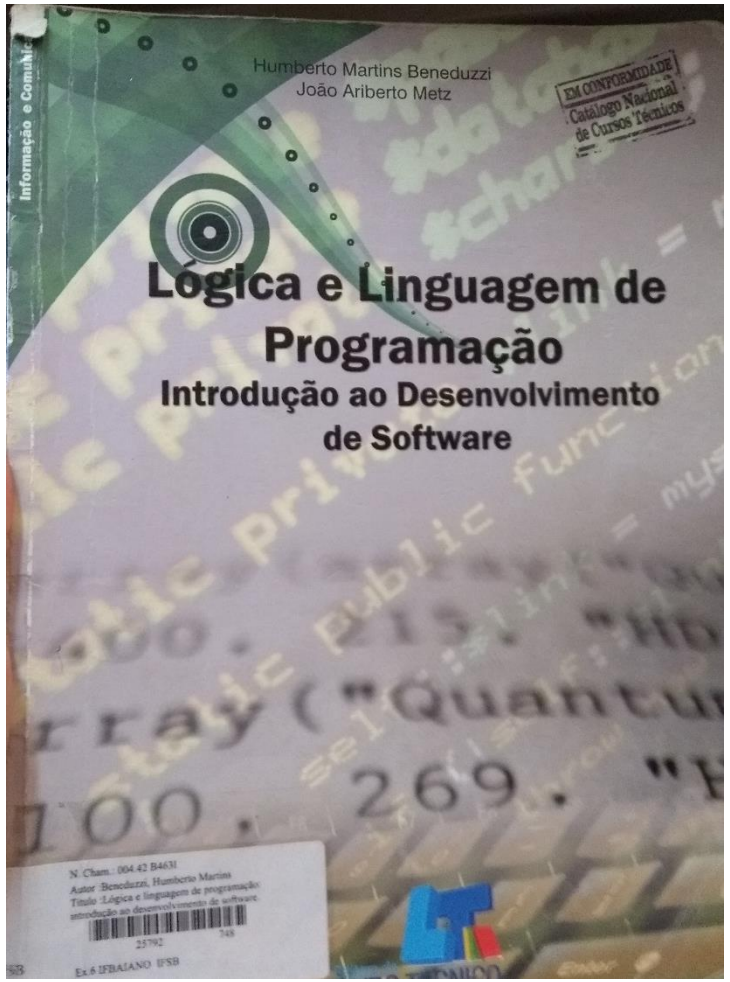
- SILVEIRA, Moraes G. da. Programação Avançada em Linux. Editora Novatec. ISBN: 8575220764
- DEITEL, H.M.; DEITEL, P.J. Java: Como Programar. 6ª ed. Pearson Education, 2005.
- DEITEL, Harvey M. C# Como programar. São Paulo: Makron Books, 2003. ISBN: 8534614598

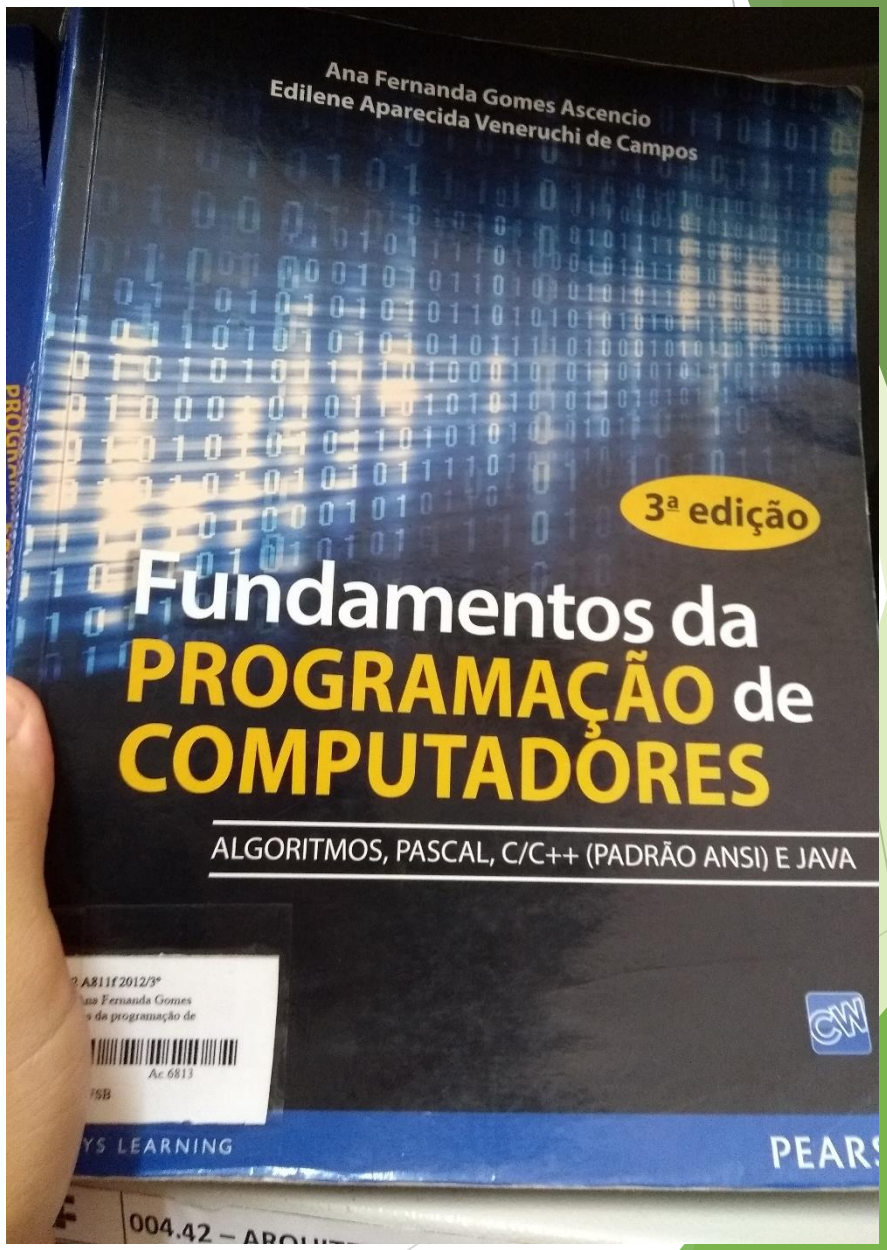
Bibliografia Complementar

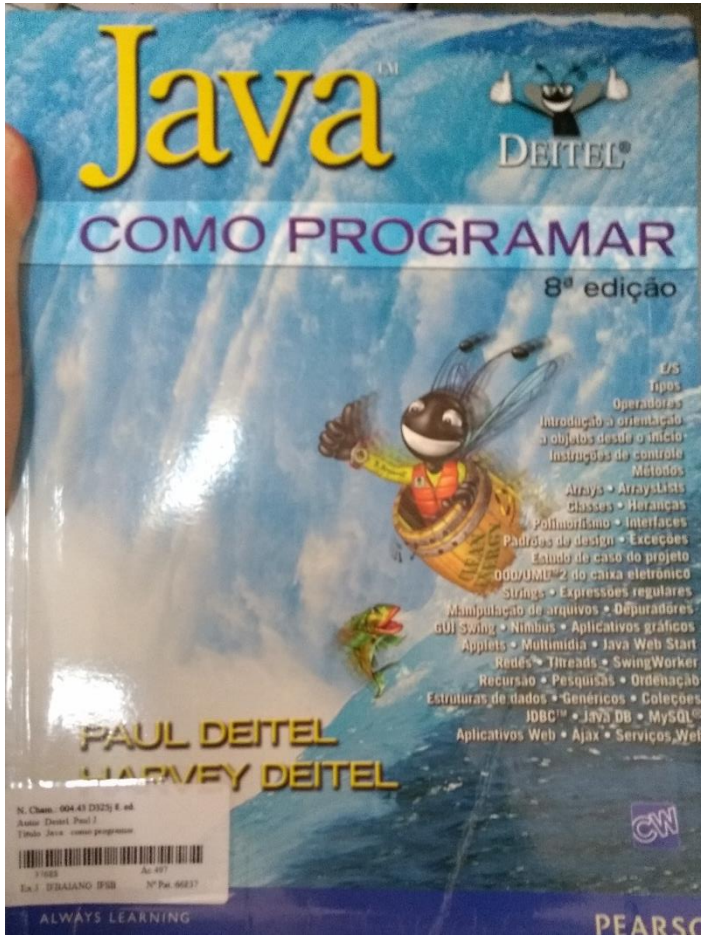
- Horstmann, Cay S. - "Big Java", Editora Bookman, 2004. (Ref. Bib. 005.133J H819b)
-
- DEITEL, Harvey M.; DEITEL, Paul J. C ++ como programar. 5. ed. Porto Alegre: Pearson, 2006. ISBN: 9788576050568

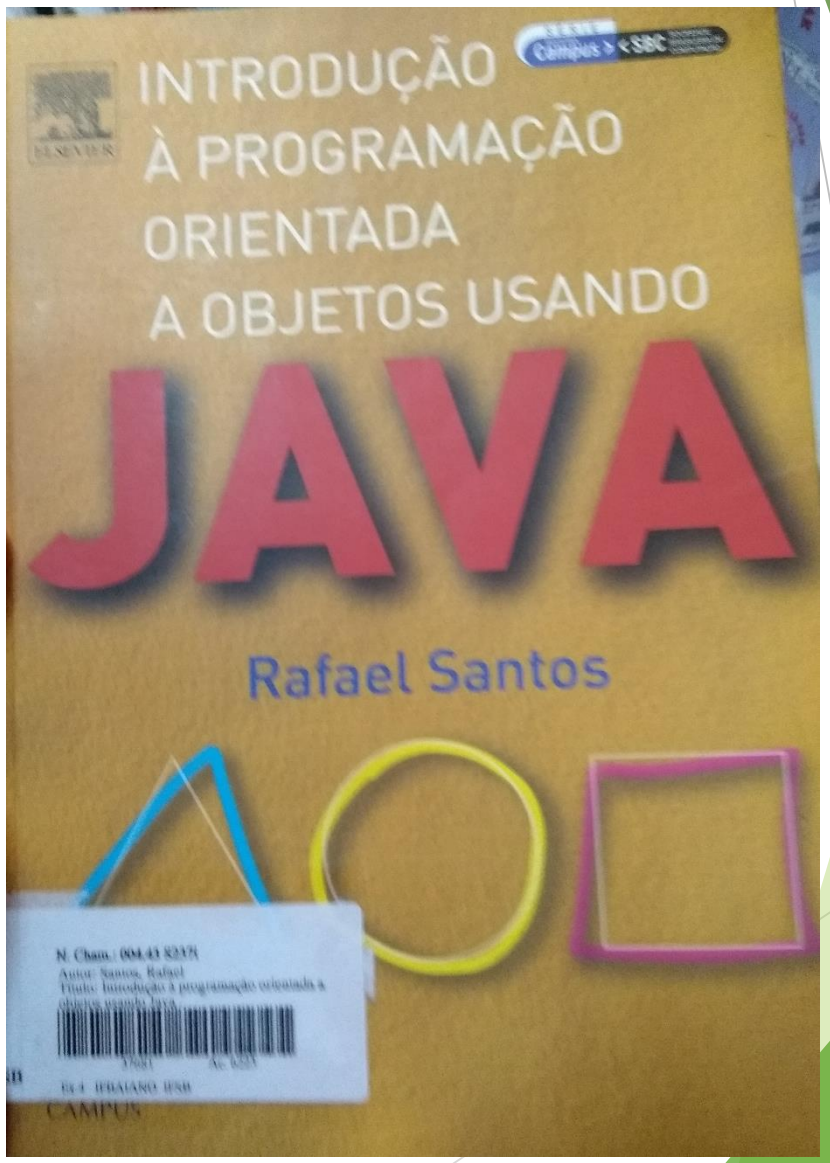

INTRODUÇÃO À PROGRAMAÇÃO
500 Algoritmos Resolvidos
Algoritmos resolvidos no cotidiano
Algoritmos resolvidos com seleção e repetição
Algoritmos resolvidos com vetores e matrizes
Algoritmos resolvidos com funções
Anita Lopes
Guto Garcia

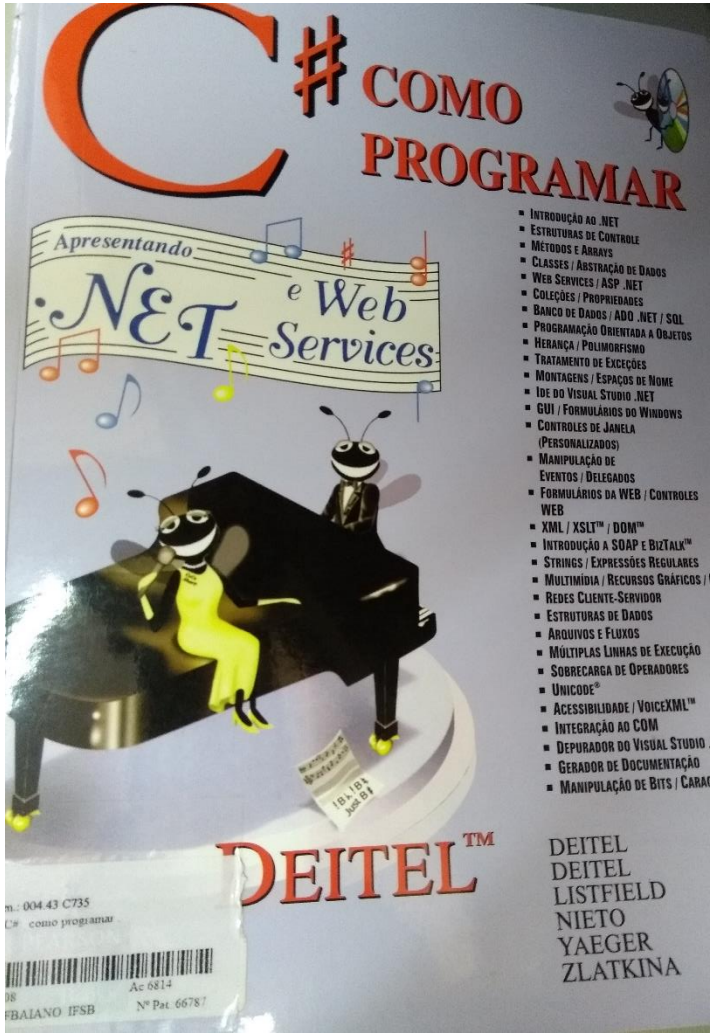

THE CLASSIC WORK
NEWLY UPDATED AND REVISED
The Art of Computer Programming
VOLUME 3
Sorting and Searching
Second Edition


Humberto Martins Beneduzzi
João Ariberto Metz
EM CONFORMIDADE
Catálogo Nacional
de Cursos Técnicos
Lógica e Linguagem de Programação
Introdução ao Desenvolvimento de Software

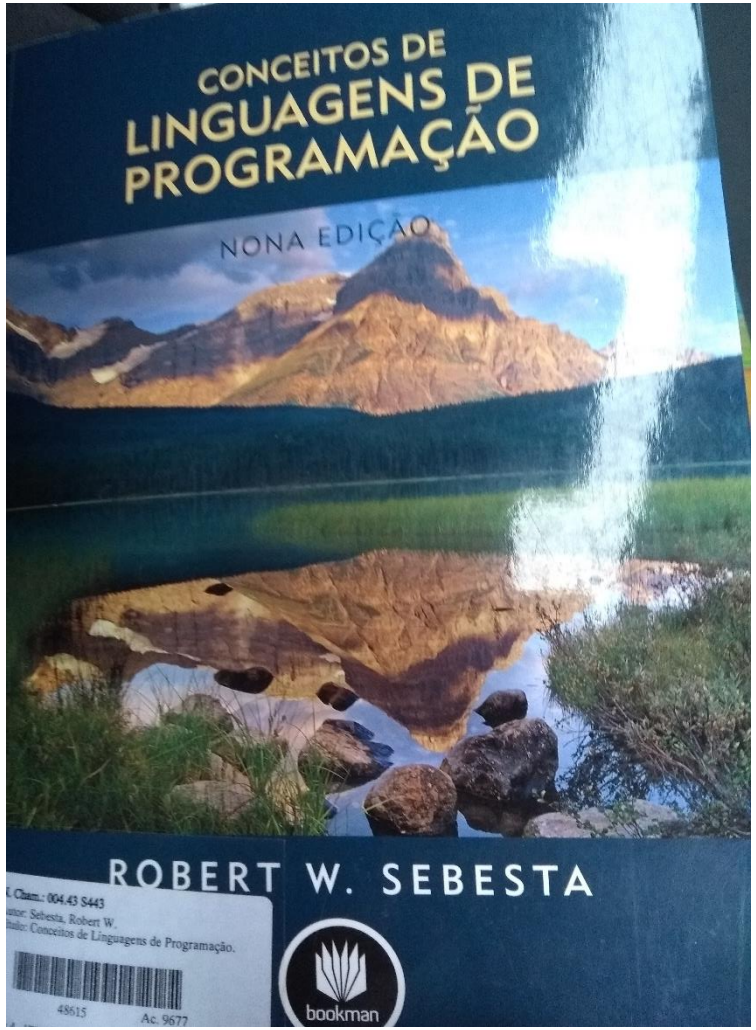

Ana Fernanda Gomes Ascencio
Edilene Aparecida Veneruchi de Campos
3ª edição
Fundamentos da PROGRAMAÇÃO de COMPUTADORES
ALGORITMOS, PASCAL, C/C++ (PADRÃO ANSI) E JAVA

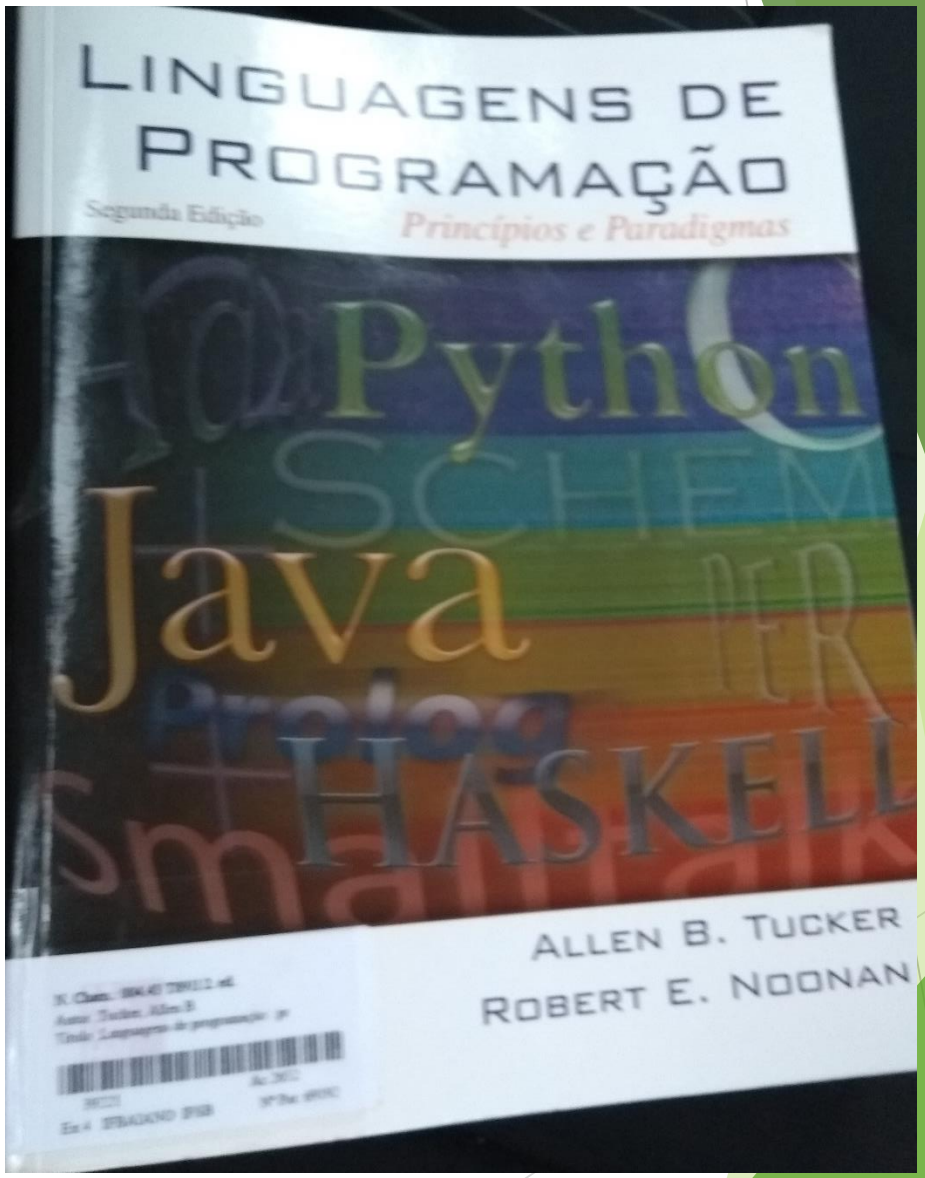

Java
DEITEL
COMO PROGRAMAR
8ª edição
E/S
Tipos
Operadores
Introdução à orientação a objetos desde o início
Instruções de controle
Métodos
Arrays • ArrayLists
Classes • Heranças
Polimorfismo • Interfaces
Padrões de design • Exceções
Estudo de caso do projeto OOD/UML™2 do caixa eletrônico
Strings • Expressões regulares
Manipulação de arquivos • Depuradores
GUI Swing • Nimbus • Aplicativos gráficos
Applets • Multimídia • Java Web Start
Redes • Threads • SwingWorker
Recursão • Pesquisas • Ordenação
Estruturas de dados • Genéricos • Coleções
JDBC™ • Java DB • MySQL®
Aplicativos Web • Ajax • Serviços Web
PAUL DEITEL
HARVEY DEITEL
ALWAYS LEARNING
PEARSON

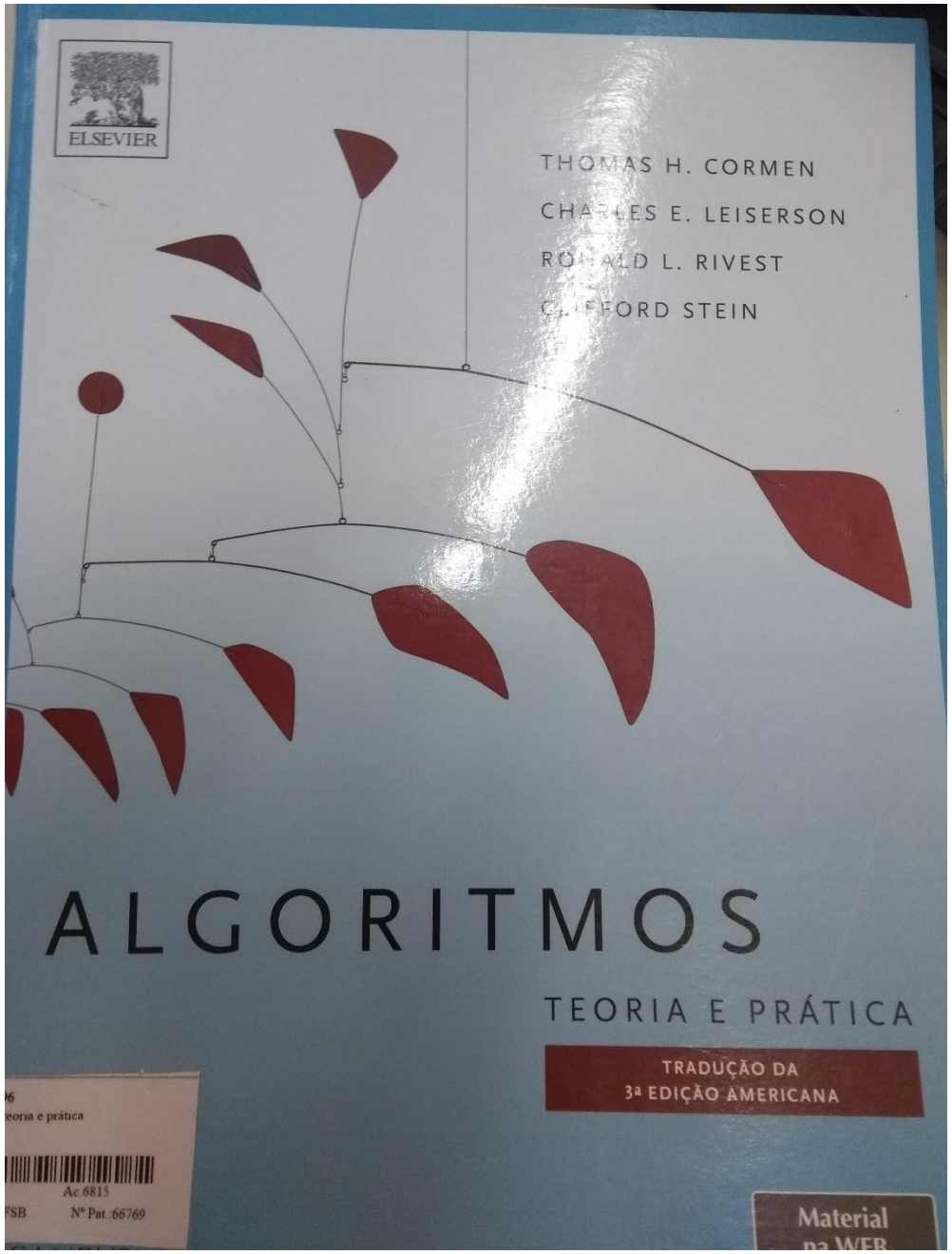

INTRODUÇÃO À PROGRAMAÇÃO ORIENTADA A OBJETOS USANDO
JAVA
Rafael Santos
CAMPUS

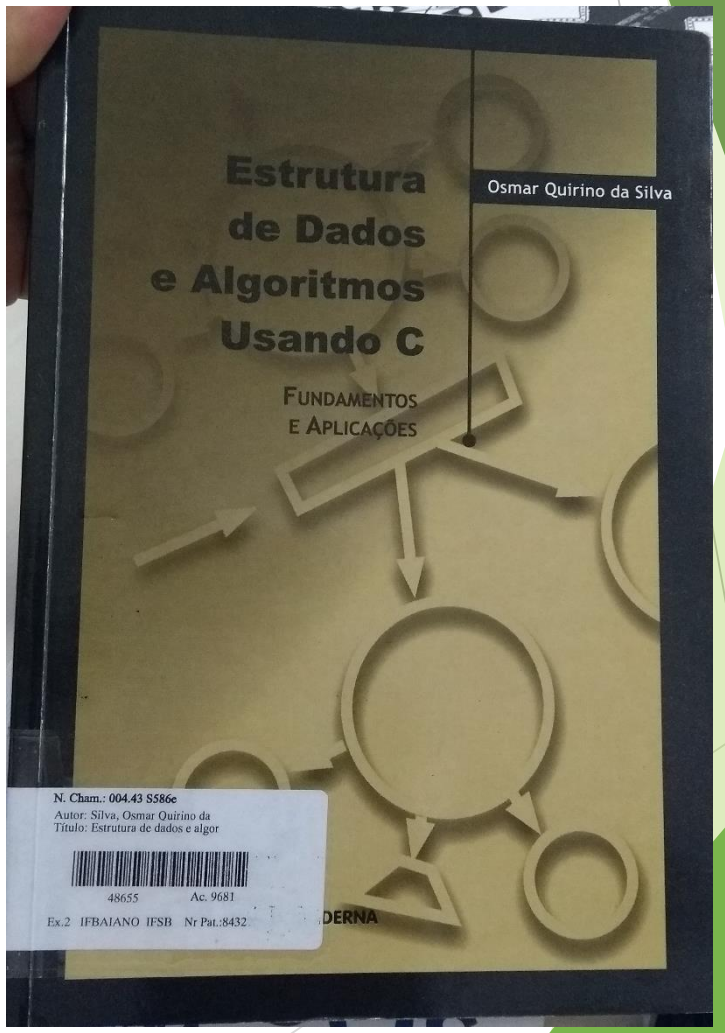

C# COMO PROGRAMAR
Apresentando .NET e Web Services
DEITEL™
DEITEL
DEITEL
LISTFIELD
NIETO
YAEGER
ZLATKINA

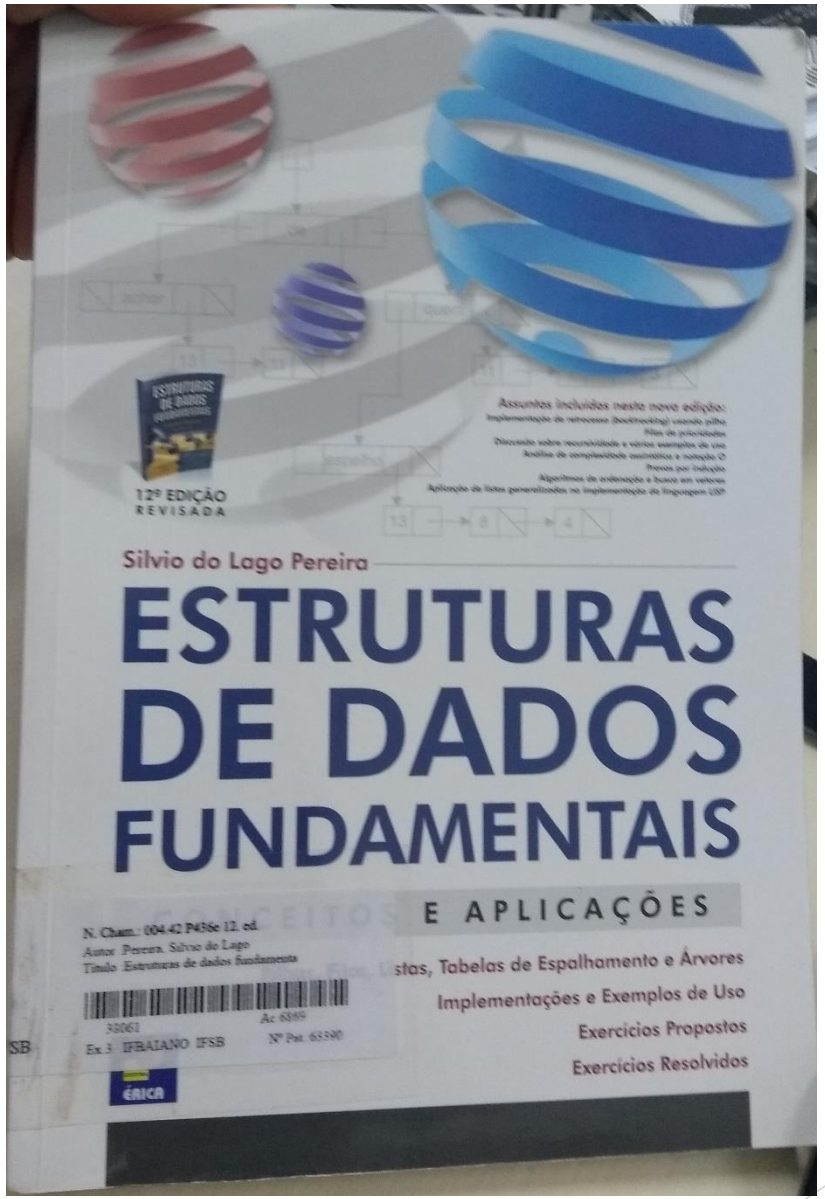

NOVA EDIÇÃO ATUALIZADA
C++ COMO PROGRAMAR
— 5ª Edição —
DEITEL
DEITEL
DEITEL®


CONCEITOS DE
LINGUAGENS DE
PROGRAMAÇÃO
NONA EDIÇÃO
ROBERT W. SEBESTA
bookman


LINGUAGENS DE
PROGRAMAÇÃO
Segunda Edição
Princípios e Paradigmas
Python
Java
HASKELL
ALLEN B. TUCKER
ROBERT E. NOONAN


ELSEVIER
THOMAS H. CORMEN
CHARLES E. LEISERSON
RONALD L. RIVEST
CLIFFORD STEIN
ALGORITMOS
TEORIA E PRÁTICA
TRADUÇÃO DA
3ª EDIÇÃO AMERICANA
Material
Ac.6815
Nº Pat.:66769


Estrutura
de Dados
e Algoritmos
Usando C
FUNDAMENTOS
E APLICAÇÕES
Osmar Quirino da Silva
N. Cham.: 004.43 S586e
Autor: Silva, Osmar Quirino da
Título: Estrutura de dados e algor
48655 Ac. 9681
Ex.2 IFBAIANO IFSB Nr Pat.:8432


12ª EDIÇÃO
REVISADA
Silvio do Lago Pereira
ESTRUTURAS
DE DADOS
FUNDAMENTAIS
E APLICAÇÕES
Implementações e Exemplos de Uso
Exercícios Propostos
Exercícios Resolvidos
ÉRICA

O que veremos hoje?

- Relembrar os velhos tempos de programação em C
- Ver os tipos de dados simples
- Funções de entrada e saída
- Declaração de variáveis locais e globais
- Declaração de variáveis homogêneas: vetores e matrizes
- Funções
  - Parâmetros
  - Retorno de funções
  - Passagens por valor e referencia
  - exercicios

Funções

```
#include<stdio.h>
void imprimirValor(int valor){          // função void não retorna valor algum
                                        // int valor, é um parâmetro da função que deseja imprimir
   printf("voce digitou o numero %i\n",valor); // utilizando o parâmetro na função
}
int media(int *n1, int *n2){            // função que retorna um valor inteiro
                                        // possui dois parâmetro, dessa vez ponteiros
   *n1 =10;                             // para alterar o valor do conteúdo do ponteiro usa-se o *
   int resultado = (*n1+*n2)/2;         // declaração de variáveis locais da função
   //printf("resultado %i\n",resultado); //parte comentada
   return resultado;                    // retorno do resultado através do return
}
```

Utilizando as funções anteriores

```
int main(){
    int numero,numero2,qualquerNumero;   //declarações de variáveis
    printf("Digite um numero\n");          // pede para o usuário digitar valores
    scanf("%i",&numero);                   // ler o primeiro valor
    imprimirValor(numero);                 // usa a função de um parâmetro só e sem retorno
    printf("Digite outro numero\n");       // pede ao usuário o segundo valor
    scanf("%i",&numero2);                  // ler o segundo valor
    imprimirValor(numero2);                //imprime usando a função
   //qualquerNumero = media(numero,numero2); // pode-se usar uma variável e receber o valor da função
   printf("%i",media(&numero,&numero2)); // pode usar a função dentro de outra
    imprimirValor(numero);                 // imprime os valores 1 e 2,
    imprimirValor(numero2);                // porém percebe-se que os ponteiros afetam os conteúdos
    return 0;
}
```

Segunda Aula

- Veremos vetores e matrizes estáticas
  - int v[4];
  - float temperaturas [5][5];
- Como acessar essas estruturas através de ponteiros dentro de uma função?
  - int media(int *vetor){};
  - float media(int **vetor){};
- Alocação dinâmica
  - int *v;
  - v = malloc (100 * sizeof (int));
  - *(v+i) = 87;