

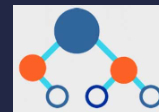


# Estrutura de Dados 1

Apresentação e Ementa

Antonio Angelo de Souza Tartaglia  
angelot@ifsp.edu.br

# Estruturas de dados 1



- Ementa

- A disciplina apresenta os conceitos e aplicações da programação modular, e as **estruturas de dados lineares** empregadas no desenvolvimento de sistemas.

# Estruturas de dados 1



- Objetivos

- Selecionar e utilizar os algoritmos e estruturas lineares para o desenvolvimento de programas de computador. Utilizar a programação modular, dominando as principais técnicas para a implementação de estruturas de dados lineares.

- Conteúdo programático

- Programação modular;
  - Procedimentos;
  - Funções;
  - Passagem de Parâmetros;
  - Concepção e implementação de programas;
- Estruturas heterogêneas (registros);
- Gerenciamento de memória;
- Recursividade;
- Tipos abstratos de dados – TAD;
  - Criação bibliotecas e módulos.
- Filas e Pilhas;
- Listas lineares e suas generalizações.

- Revisão Linguagem C;
  - Programação Modular;
  - *Strings* – Cadeia de caracteres;
  - *structs* - Estruturas Heterogêneas;
  - Ponteiros;
  - Manipulação de Arquivos;
  - Recursividade;
  - Alocação de Memória;
- 
- Tipo Abstrato de Dados (TAD);
  - Lista Sequencial Estática;
  - Lista Ligada Dinâmica;
  - Filas e Pilhas Estáticas e Dinâmicas;
  - Fila de Prioridades;
  - Lista Duplamente Ligada;
  - Lista Ligada Circular.

# Estruturas de dados 1



*Brian W. Kernighan*



*Dennis Ritchie*

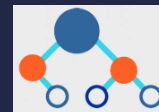
- A Linguagem C

- Criada em 1972, nos laboratórios Bell, por Brian W. Kernighan e Dennis Ritchie;
- Estruturalmente simples e de grande portabilidade;
- É uma Linguagem Procedural – decompõe problemas complexos em módulos ou **blocos\*** mais simples;
- Afinal, C é de alto ou baixo nível? – combina elementos de alto e baixo nível. Permite manipular bits, bytes e endereços de memória (baixo nível), possibilitando assim a programação direta ao nível do microprocessador, ao mesmo tempo que possui comandos próximos da linguagem humana;
- Permite a inserção de trechos de código em linguagem Assembly – para situações onde o tempo é uma questão crítica.



**Blocos:** Conjunto de instruções que estão ligadas logicamente.





- A Linguagem C

- A que área de desenvolvimento se destina?
- Nenhuma em particular.
- Ela é aquilo que se denomina “de uso geral”, e esta é uma de suas grandes vantagens, pois adapta-se ao desenvolvimento de qualquer projeto, tais como:
  - Sistemas Operacionais;
  - Interfaces Gráficas;
  - Drivers para componentes de hardware;
  - Processamento de Registros;
  - E, por incrível que pareça C também é utilizada para escrever compiladores de outras linguagens



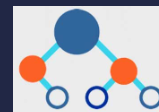
- A Linguagem C
- É uma linguagem extremamente potente e flexível:
  - Rapidez – Consegue obter performances semelhantes às obtidas pelo Assembly - que utiliza mnemônicos próximos a linguagem de máquina – e o faz utilizando instruções de **alto nível**;
  - Simples – Sintaxe extremamente simples, número de palavras reservadas, tipos de dados básicos e de operadores diminutos;
  - Poderosa - Apesar de sua sintaxe ser mais próxima à humana, é possível manipular a memória diretamente utilizando o endereço de qualquer objeto (seja uma variável ou uma função), atuando diretamente na memória sem qualquer tipo de restrição, o que aumenta a flexibilidade da linguagem, seu poder e também o perigo;



- A Linguagem C

- Conta com Bibliotecas Poderosas – O fato de possuir um conjunto de instruções reduzido, sugere que as capacidades de C são reduzidas, e na realidade são mesmo! A maior parte das funcionalidades da linguagem é obtida pela utilização de funções que existem em bibliotecas adicionais, e que realizam todo o tipo de tarefas;
- Portável – Seu código fonte pode ser portado e compilado em diversos sistemas operacionais, por ser padronizada em 1989 pelo comitê ANSI (*American National Standards Institute*), que é o órgão regulamentador e define as características de qualquer compilador.
- Atualmente a versão padronizada e distribuída é de 2017, mas já está em fase final de validação uma versão que deve sair ainda este ano, o documento que a descreve pode ser encontrado em:  
<https://www.open-std.org/jtc1/sc22/wg14/www/docs/n3088.pdf>
- Popular – Internacionalmente conhecida e utilizada, e muito bem documentada;
- Modular – Permite o desenvolvimento modular de aplicações, facilitando a separação de projetos em módulos distintos e independentes, recorrendo a funções específicas dentro de cada módulo;





- A Linguagem C - evolução

- A Evolução das linguagens fez com que C evoluísse no sentido das **Linguagens Orientadas a Objetos**, dando origem a uma nova linguagem, que se tornou um super conjunto de C: a C++, que mantém a mesma sintaxe da Linguagem C, e a expande, introduzindo um conjunto adicional de características inerentes à **OO**:

- Encapsulamento;
- Hereditariedade;
- Polimorfismo;
- Sobrecarga;
- Etc.



# Estruturas de dados 1



- A Linguagem C



- A Linguagem C tem influenciado direta ou indiretamente, muitas linguagens desenvolvidas posteriormente:

- C++;
- Java;
- C#;
- PHP;
- JavaScript.
- Rust
- Python



- Todas as mencionadas acima, e várias outras, combinam a sintaxe de declaração e a sintaxe de expressão da linguagem C como sistemas de tipos, modelos de dados, etc.

# Estruturas de dados 1



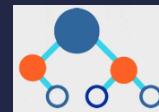
- Principais Razões para o aprendizado e utilização da Linguagem C:
  - Ela é uma linguagem relativamente simples de aprender e até de usar;
  - Não há ferramenta melhor para resolver alguns tipos de problemas;
  - Há muita coisa feita em C que precisa de compatibilidade (sistemas legados), é a língua universal da computação, comparativamente, é como se fosse o inglês;
  - Está disponível virtualmente em todas plataformas;
  - Muito poderosa e flexível, faz o serviço de forma muito eficiente;
  - Acessa o baixo nível, se comunica bem com tudo, e te dá todo controle **sobre tudo**, sem interferências.

# Estruturas de dados 1



- Onde é utilizada:
  - Tudo que tem poucos recursos precisam de controle absoluto (embarcados, alguns IoT – Internet of Things);
  - Sistemas operacionais que não podem ter altos custos computacionais e precisam acessar diretamente o hardware;
  - Softwares que devem responder para outros e em grande volume (por exemplo drivers de componentes de hardware), e que não podem suportar desperdícios de recursos (processamento e memória);

# Estruturas de dados 1



- Avaliações

- EX – Exercícios propostos em cada aula + Projeto de Extensão, compondo 20% da nota final;
- P1 – Avaliação escrita + avaliação prática desenvolvida em sala de aula a partir de requisitos propostos, com a média das duas compondo 40% da nota final;
- PF - Projeto final – Projeto prático de elaboração de software específico, compondo 40% da nota final;
- Obs.: A depender do desempenho da turma, é possível a aplicação de mais uma avaliação escrita, que então comporá também os 40% do Projeto Final (média entre os dois).

- **Média final** =  $0,2 * EX + 0,4 * PJ + 0,4 * TF$

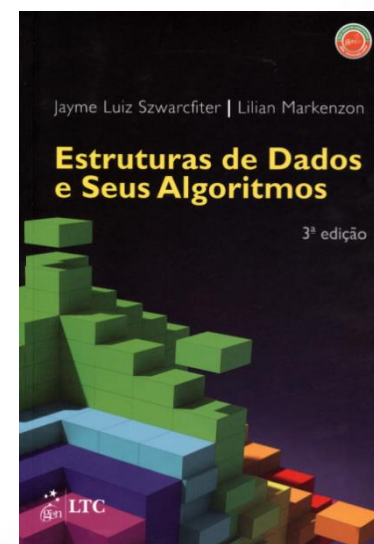
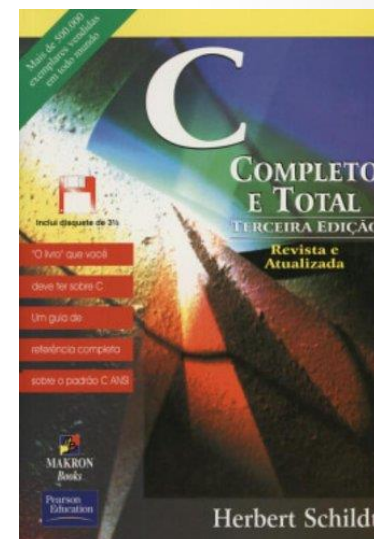
# Estruturas de dados 1



```
if(frequencia >= 75%){  
    if(Nota Final >= 6.00){  
        printf("Você foi aprovado!!");  
    }else{  
        printf("ihhh, você não foi aprovado...");  
  
        if(Nota Final >= 4.00){  
            printf("Você está apto à prestar o EXAME.");  
        }else{  
            printf("Você não atingiu a nota para exame, está reprovado.");  
        }  
    }  
}else{  
    printf("Frequencia insuficiente, você está reprovado!");  
}
```

# Estruturas de dados 1

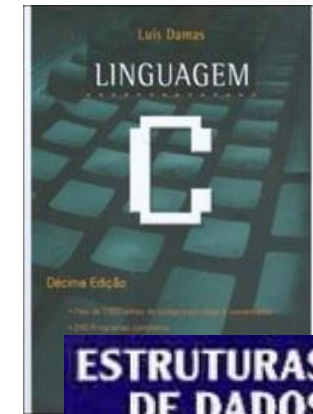
- Bibliografia Básica
- BAKES, André. **Estrutura de dados descomplicada**: em Linguagem C. Rio de Janeiro: Elsevier, 2016. 420p.
- SCHILDT, Herbert. **C completo e total**. 3ª. Ed. São Paulo: Pearson Makron Books, 1997.
- SZWARCFITER, J. L.; MARKENZON, L., Estruturas de dados e seus algoritmos. 3.ed. Rio de Janeiro: LTC, 2010.



# Estruturas de dados 1

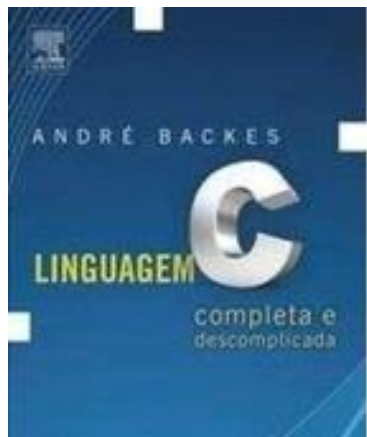
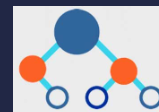


- Bibliografia Complementar
- BACKES, André. **Linguagem C: completa e descomplicada**. Rio de Janeiro: Elsevier, 2013. 400p.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R.; STEIN, C. **Algoritmos: teoria e prática**. Rio de Janeiro: Campus, 2002.
- DAMAS, Luis. **Linguagem C**. 10ª ed. Tradução João Araújo Ribeiro, Orlando Bernardo Filho. Rio de Janeiro: LTC, 2015. 412p.
- DROZDEK, A. **Estrutura de dados e algoritmos em C++**. São Paulo: Pioneira Thomson Learning, 2002.
- TANENBAUM, A.; LANGSAM, Y.; AUGENSTEIN, M. **Estruturas de dados usando C**. São Paulo: Pearson, 1995.





# Estruturas de dados 1



## Material complementar:

- Linguagem C completa e Descomplicada  
autor: André Backes.
- Estrutura de Dados Descomplicada em Linguagem C
- Autor: André Backes.

- Site:  
<https://programacaodescomplicada.wordpress.com/indice/linguagem-c/>
- <https://programacaodescomplicada.wordpress.com/indice/estrutura-de-dados/>

