

ALGORITMOS E ESTRUTURA DE DADOS I



Interdisciplinar em Tecnologia da Informação - UFERSA

Professora: Dra. Rosana Rego

Quem são os alunos?



EMENTA

□ Parte 1

4

- Ponteiros (Endereço, indireção de ponteiros, ponteiro nulo)
- Alocação dinâmica (Alocação de vetores e matrizes)
- Tipos Estruturados
 - Registros
 - Definição de “novos” tipos
 - Vetores de estruturas
 - Vetores de ponteiros para estruturas
 - Uniões e enumerações

□ Parte 2

- Arquivos
- Algoritmos de busca e ordenação
- Tipos abstratos de dados

EMENTA

□ Parte 3

4

- Listas encadeadas
- Listas encadeadas e seus algoritmos
- Listas Circulares
- Listas duplamente encadeadas
- Listas de tipos estruturados
- Pilhas
- Interface do tipo pilha
- Implementações de pilha
- Filas
- Interface do tipo fila
- Implementação de fila
- Fila dupla

AVALIAÇÕES

- Parte 1
 - Prova: 24/02
 - Projetos/Listas;
- Parte 2
 - Prova: 31/03
 - Projetos/Listas;
- Parte 3
 - Projetos/Listas;

Softwares – Programas de computador



Visual Studio Code:

<https://code.visualstudio.com/download>



DevC++:

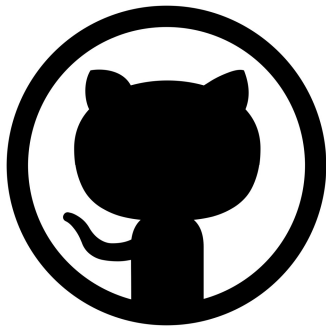
<https://sourceforge.net/projects/orwelldvcpp/>



CodeBlocks:

<http://www.codeblocks.org/downloads>

Repositório

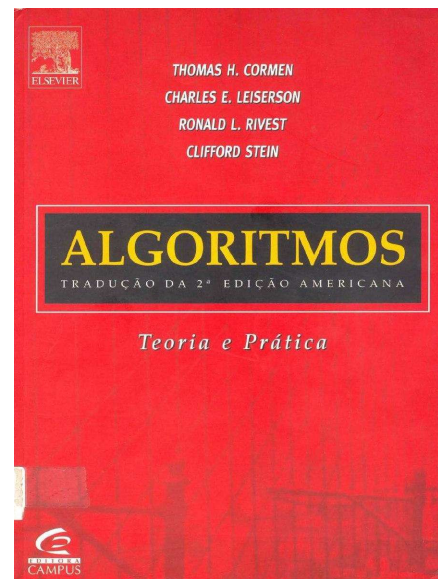
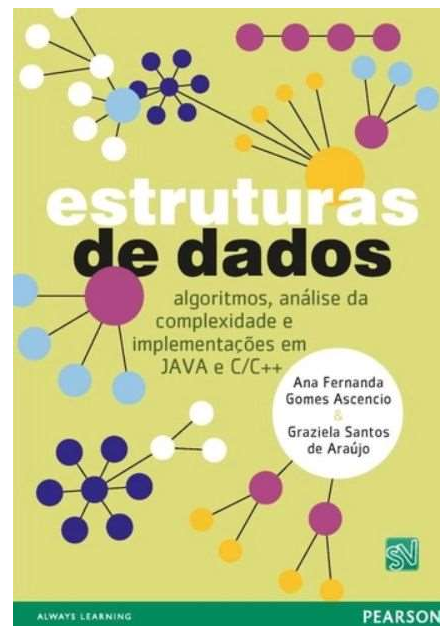
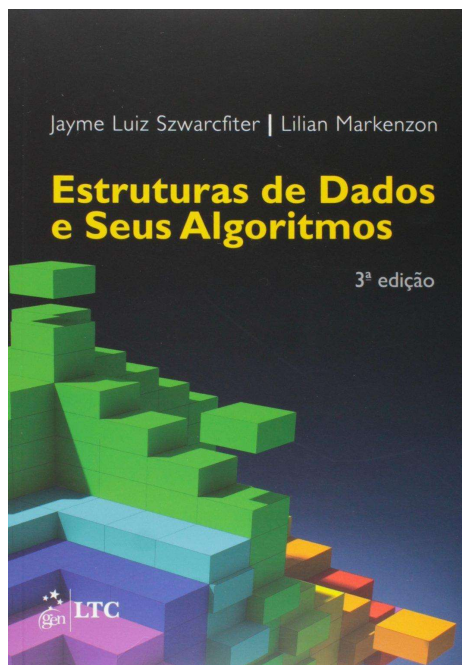


<https://github.com/roscibely/algorithms-and-data-structure>



<https://github.com/roscibely/data-structure-with-python>

Livros



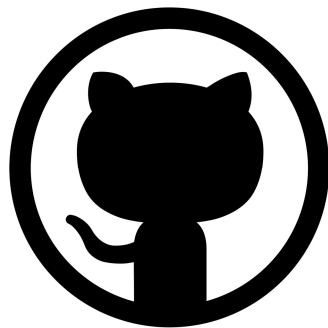
Bibliografia Básica:

1. ASCENCIO, A. F. G. Estrutura de Dados. São Paulo: Pearson, 2011. 448p.
2. SZWARCFITER, J. L.; MARKENZON L. Estruturas de Dados e Seus Algoritmos. 3ª ed. Rio de Janeiro: LTC, 2010. 318p.
3. ZIVIANI, N. Projeto de Algoritmos com Implementações em Java e C++. Thomson Learning, 2006. 642p.

Bibliografia Complementar:

1. TOSCANI, L. V.; VELOSO, P. A. S. Complexidade de Algoritmos – Vol. 13. 3ª ed. Porto Alegre: Bookman, 2012. 280p. (Série de livros didáticos informática UFRGS).
2. LOPES, A.; GARCIA, G. Introdução a Programação. Rio de Janeiro: Campus, 2002. 488p.
3. DEITEL, P.; DEITEL, H. C: Como Programar. 6ª ed. São Paulo: Pearson, 2007. 848p.
4. TENENBAUM, A. M.; LANGSAM, Y.; AUGENSTEIN, M. J. Estruturas de dados Usando C. São Paulo: Makron Books, 1995. 904p;
5. MIZRAHI, V. V. Treinamento em Linguagem C. 2ª ed. São Paulo: Prentice-Hall, 2008. 432p

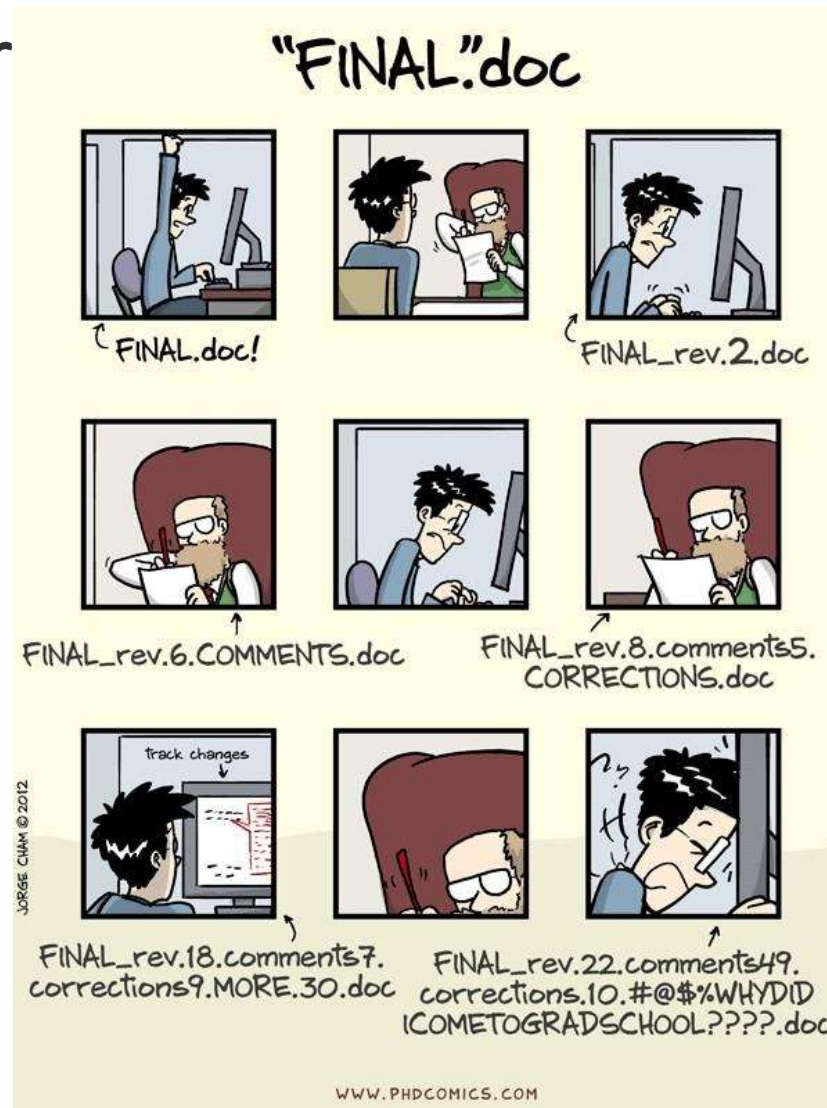
CONTROLE DE VERSÃO DE SOFTWARES



Armazenando seu projeto

- Como armazenar versões?
 - Não se preocupar
 - Boa sorte!
 - Acumular arquivos ZIP numerados (ou com data da versão)
 - Como verificar as diferenças no código (Diff)
 - Sistema de Gerenciamento de Versões
 - Facilidade para verificar as mudanças no código entre versões
 - Facilidade em restaurar versões para testes

Por que usar um Sistema de Gerenciamento de Versões



Por que usar um Sistema de Gerenciamento de Versões?

- Histórico das alterações
 - Responsável pelas alterações (quem quebrou o build)
- Facilidade para voltar atrás e restaurar o código
- Não precisa se preocupar em estragar um código que funciona
- Permite juntar alterações feitas por diversos desenvolvedores

Sistema de Gerenciamento de Versões

- Soluções Comerciais
 - Microsoft Visual SourceSafe (VSS)
 - Rational ClearCase
 - Borland StarTeam
- Soluções Open-source
 - Concurrent Version System (CVS)
 - Subversion (SVN)
 - Git
 - Mercurial

Sobre o Git

- Git é um Sistema open-source de gerenciamento de versões
- Foi desenvolvido pelo Linus Torvalds (desenvolvedor do Linux)
 - Criado em 2005 para gerenciar o desenvolvimento dos códigos do Linux
- Pode ser usado para controlar versões de qualquer formato (geralmente arquivos de texto)
 - Códigos fonte
 - Projetos de análise de dados
 - Manuscritos
 - Websites
 - Apresentações
 - Etc...

Por que usar o Git?

- É rápido. O foco do sistema é a velocidade.
- Não é necessário ter acesso ao direto servidor, basta usar sua conta git.
- Muito bom em gerenciar e unificar modificações simultâneas ao mesmo arquivo
- Tem se tornado o principal protocolo de gerenciamento de versões.



Onde usar?



- <https://bitbucket.org/>
 - Git ou Mercurial
 - Repositórios privados gratuitos



GitHub

- <https://github.com/>
 - Git
 - Repositórios privados ou públicos

Sobre o GitHub

- É um site para repositórios Git.
- Possui interface para explorar repositórios git.
- Verdadeiro open source
 - Permite acesso fácil ao código de diversos desenvolvedores
- É como o Facebook para programadores

Por que usar o GitHub?

- É otimizado e existe uma contínua preocupação com os servidores para os aspectos do git.
- Interface gráfica para os repositórios
 - Permite a exploração do código e seus históricos.
 - Permite registrar e monitorar bugs ou tarefas
- Auxilia
 - ... no aprendizado através do projetos de outros usuários.
 - ... na verificação do que os outros estão fazendo.
 - ... na contribuição com o código de outros.
- Facilita o processo colaborativo
 - “Existe um problema na sua documentação” vs “Aqui está uma correção para a sua documentação”.

Ferramentas



- Git for Desktop
 - <https://desktop.github.com/>



- Tortoise Git
 - <https://tortoisegit.org/>

Conclusões

- Gerenciar as versões do seu código auxilia na colaboração entre diversos desenvolvedores, no lançamento de versões do projeto, na verificação de erros e na recuperação de versões corretas.
- Criar um projeto open-source significa que todos poderão verificar os erros do seu código.
- Utilizar um controle de versão quer dizer que todos poderão verificar todos os erros do seu código. (mesmo os antigos que foram corrigidos)

Curso Complementar

- Git e controle de versão: 🖐️

<https://www.dataquest.io/course/git-and-vcs/>

Tempo estimado: 5h