

# Estruturas de Dados 1

## Apresentação da Disciplina

Mailson de Queiroz Proença

# Apresentação do professor

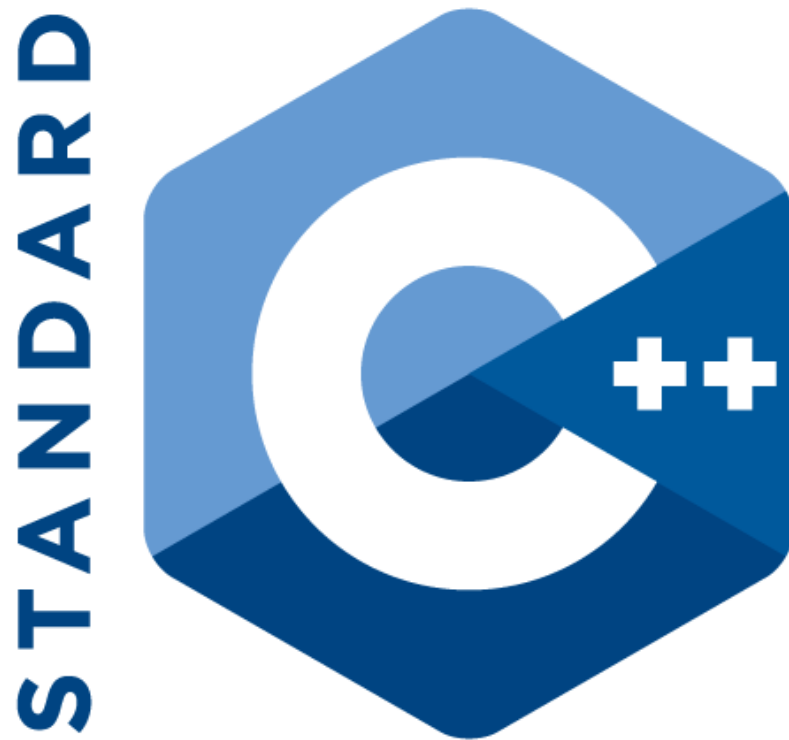
- Mailson de Queiroz Proença
  - Formação Acadêmica:
    - Bacharel em Sistemas de Informação - FESP/UEMG (2013);
    - Especialista em Plataforma de Desenvolvimento Web - Claretiano (2014);
    - Mestrando em Ciências da Computação - UFSCar.

# O que vamos aprender

- Mais programação, ou seja, mais códigos fontes;
- Conceitos avançados de programação;
- Como funciona registros, funções e arquivos;

# O que vamos aprender

- Qual a linguagem?
- Vamos estudar a linguagem de programação C++.



# O que vamos aprender

## Linguagem C++:

Atualmente é a 3ª linguagem de programação mais utilizada no mundo:

Jul 2016	Jul 2015	Change	Programming Language	Ratings	Change
1	1		Java	19.804%	+2.08%
2	2		C	12.238%	-3.91%
3	3		C++	6.311%	-2.33%
4	5	⬆	Python	4.166%	-0.09%
5	4	⬇	C#	3.920%	-1.73%
6	7	⬆	PHP	3.272%	+0.38%
7	9	⬆	JavaScript	2.643%	+0.45%
8	8		Visual Basic .NET	2.517%	+0.09%
9	11	⬆	Perl	2.428%	+0.62%
10	12	⬆	Assembly language	2.281%	+0.75%

Fonte: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

# O que vamos aprender

Qual programa utilizar:

CodeBlocks



Dev C++



Falcon C++;



# Habilidades e Competências

- Identificar oportunidades de mudanças e projetar soluções usando tecnologias da informação nas organizações;
- Modelar e implementar soluções de Tecnologia de Informação em variados domínios de aplicação;
- Identificar e projetar soluções de alto nível e opções de fornecimento de serviços, realizando estudos de viabilidade com múltiplos critérios de decisão.

# Objetivos

- Dominar a programação estruturada e as principais estruturas de dados utilizadas na computação;
- Ser capaz de escolher e usar corretamente as estruturas de dados mais adequadas para programar de forma otimizada.



# Ementa

- Tipos de dados avançados: conjuntos, registros, vetores de registros.
- Modularização de programas (subprogramas), procedimentos e funções. Passagem de Parâmetros.
- Conceitos de Recursividade.
- Programação com arquivos.
- Depuração e documentação de programas.

# Estratégias de Ensino

- Aulas expositivas de conceitos em sala;
- Criação participativa de rotinas para tarefas com estruturas de dados estudadas;
- Prática de desenvolvimento de soluções em ambiente de desenvolvimento em aulas no laboratório de informática;
- Uso de listas de exercícios para complementação e fixação do aprendizado;
- Realização de trabalhos práticos avaliativos;
- Provas avaliativas;
- Disponibilização de material através de ambiente de Ensino a Distância como apoio ao ensino presencial.

# Instrumentos de Avaliação (1/3)

- A distribuição dos pontos no semestre será feita da seguinte maneira:
  - Prova 1 (P1): 40 pontos
  - Prova 2 (P2): 40 pontos
  - Trabalho(s) prático(s) (Trab): 20 pontos

# Instrumentos de Avaliação (2/3)

- Sobre as provas:
  - Cada prova tem caráter de avaliação individual.
- Sobre os trabalhos práticos:
  - Os temas serão divulgados na Intranet;
  - Os trabalhos práticos poderão ser desenvolvidos em dupla;
  - Havendo cópia entre duplas, ambas perdem a nota de todo trabalho;
  - O(s) autor(es) poderão ser chamados, individualmente ou em grupo, para dar explicações sobre o trabalho a fim de comprovar sua autoria;
  - A entrega dos trabalhos práticos deverão seguir os moldes divulgados na Intranet.

# Instrumentos de Avaliação (3/3)

- Síntese dos critérios para aprovação nas unidades curriculares por semestre:
  - Se Frequência  $< 75\%$ : Reprovado
  - Se Frequência  $\geq 75\%$ : Verifica Média
  - Média =  $(P1 + P2 + Trab)$
  - Se Média  $\geq 60$  então Aprovado
  - Se Média  $< 40$  então Reprovado
  - Se Média  $\geq 40$  E Média  $< 60$  então Prova Final (PF)
    - Se PF  $\geq 60$ : Aprovado
    - Senão: Reprovado

# Bibliografia Básica

- ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de. Fundamentos da programação de computadores: algoritmos, pascal, C/C++ padrão ansi e java . 3. ed. São Paulo, SP : Pearson Education do Brasil, [2012].
- FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. Lógica de programação: a construção de algoritmos e estruturas de dados . 3. ed., 6. reimpr. São Paulo, SP : Pearson Prentice Hall, [2010].
- OLIVEIRA, Jayr Figueiredo de; MANZANO, José Augusto N. G.. Algoritmos: lógica para o desenvolvimento de programação de computadores . 26. ed., rev. São Paulo, SP : Ed. Érica, 2012.

# Bibliografia Complementar

- DEITEL, Harvey M. et al. **C++ como programar**. São Paulo: Pearson Makron Books, 2006.
- TREMBLAY, Jean Paul; BUNT, R. B. **Ciências dos Computadores: uma abordagem algorítmica**. São Paulo: McGraw-Hill, 1983.
- WIRTH, Niklaus. **Algoritmos e Estruturas de Dados**. Rio de Janeiro: Prentice-Hall, 1999.

# Avisos



- Horário de Aula: 19h10 - 20h40 e 21h00 - 22h30.
- Celulares no silencioso, atendam fora da sala caso precisem.
- Controlem as conversas paralelas !!!
- Todos os programas e exemplos em sala de aula foram testados.
- Estarei a disposição para tirar dúvidas !!!!
- Participação de todos na aula.
- É uma disciplina complexa, por isso mantenha a atenção.
- Para aprender corretamente a disciplina, devemos praticar bastante tanto no laboratório, sala de aula e em atividades extracurriculares.





Prof. Espec. Mailson de Queiroz Proença

E-mail: [mailson.proenca@uemg.br](mailto:mailson.proenca@uemg.br)

**Bom semestre para todos!**