



Tópicos Especiais em Engenharia de Software:  
Técnicas Avançadas de Programação

# Plano de Ensino

Prof. Leandro M. Zatesko  
1º semestre de 2016

## 1 Identificação

**Curso:** Ciência da Computação

**Turno/Fase:** Matutino/7ª

**Componente curricular:** Tópicos Especiais em Engenharia de Software (GEX389)

**Turma:** 13250

**Créditos:** 4

**Carga horária (em horas):** 60

**Carga horária (em horas-aula):** 72

**Professor:** Leandro Miranda Zatesko

Atendimento ao estudante*	
Local	Bloco dos professores, sala 220
Horário	Terças-feiras, das 14:30 às 17:30

Horários dos encontros	
Quartas-feiras	das 8:20 às 10:00**
Quintas-feiras	das 10:10 às 11:50

\* Outros horários podem ser agendados pelo fórum.

\*\* O horário dos encontros destacados em **negrito** no Cronograma (cf. Seção 5) é das 7:30 às 10:00.

## 2 Objetivo geral do Curso

O curso tem por objetivo a formação integral de novos cientistas e profissionais da computação, os quais deverão possuir conhecimentos técnicos e científicos e serem capazes de aplicar estes conhecimentos, de forma inovadora e transformadora, nas diferentes áreas de conhecimento da Computação. Adicionalmente, os egressos do curso deverão ser capazes de adaptar-se às constantes mudanças tecnológicas e sociais, e ter uma formação ao mesmo tempo cidadã, interdisciplinar e profissional.

## 3 Ementa

Programação competitiva. Estruturas de dados avançadas. Algoritmos gulosos. Algoritmos de divisão e conquista. Programação dinâmica. Modelagem de problemas com grafos. Problemas combinatoriais. Problemas de Teoria dos Números. String matching. Geometria computacional.

## 4 Objetivos

### 4.1 Geral

Estudar técnicas avançadas de programação para a resolução de problemas complexos, praticando a implementação das técnicas em diversos exercícios e comparando analiticamente o resultado com os algoritmos de força bruta.



## 4.2 Específicos

- Aprimorar a criatividade e as habilidades necessárias para competições de programação.
- Fomentar o espírito de trabalho em equipe e o interesse por Programação Competitiva.
- Motivar a formação de times qualificados para a Maratona de Programação.
- Desenvolver as habilidades acadêmicas de exposição de resultados e argumentação.

## 5 Cronograma e conteúdos programáticos

Data	Programa	
<b>Março</b>		
2	Apresentação do Plano de Ensino Introdução à Programação Competitiva	Publicação da Lista de Exercícios 1
3	Problemas <i>Ad Hoc</i>	
9	Estruturas de dados implementadas na STL do C++: vector, priority_queue, set e map	Publicação da Lista de Exercícios 2
10	Estruturas de dados implementadas na STL do C++: vector, priority_queue, set e map	
16	Estruturas de dados: MUF, BIT e árvores de segmento	Publicação da Lista de Exercícios 3
17	Estruturas de dados: MUF, BIT e árvores de segmento	
23	Paradigmas de resolução de problemas: D&C, <i>backtracking</i> e algoritmos gulosos	Publicação da Lista de Exercícios 4
24	Paradigmas de resolução de problemas: D&C, <i>backtracking</i> e algoritmos gulosos	
30	Paradigmas de resolução de problemas: Programação Dinâmica	Publicação da Lista de Exercícios 5
31	Paradigmas de resolução de problemas: Programação Dinâmica	
<b>Abril</b>		
6	Grafos: DFS, BFS e aplicações	Publicação da Lista de Exercícios 6
7	Grafos: DFS, BFS e aplicações	
13	Grafos: Algoritmo de Tarjan Grafos: outras aplicações de buscas em grafos	Publicação da Lista de Exercícios 7
14	Grafos: Algoritmo de Tarjan Grafos: outras aplicações de buscas em grafos	
20	<b>Grafos: árvores geradoras mínimas</b>	Publicação da Lista de Exercícios 8
21	Tiradentes (feriado nacional)	
27	Grafos: caminhos mínimos	Publicação da Lista de Exercícios 9
28	Grafos: caminhos mínimos	

Maio		
4	<b>Grafos: fluxo e aplicações</b>	Publicação da Lista de Exercícios 10
5	Grafos: fluxo e aplicações	
11	Combinatória	Publicação da Lista de Exercícios 11
12	Combinatória	
18	Teoria dos Números	Publicação da Lista de Exercícios 12
19	Teoria dos Números	
25	<b>Strings: string matching e string processing</b>	Publicação da Lista de Exercícios 13
26	<i>Corpus Christi</i> (feriado nacional)	
Junho		
1	<b>Strings: suffix array e aplicações</b>	Publicação da Lista de Exercícios 14
2	Strings: suffix array e aplicações	
8	Geometria Computacional: objetos básicos	Publicação da Lista de Exercícios 15
9	Geometria Computacional: objetos básicos	
15	Geometria Computacional: algoritmos básicos	Publicação da Lista de Exercícios 16
16	Geometria Computacional: algoritmos básicos	
22	Tópicos avançados	Publicação da Lista de Exercícios 17
23	Tópicos avançados	
29	Tópicos avançados	Publicação da Lista de Exercícios 18
30	Tópicos avançados	
Julho		
4	Exame Final	

## 6 Procedimentos metodológicos

### Encontros acadêmicos em sala de aula

Todos os encontros acadêmicos ocorrerão no Laboratório 405B. O conteúdo será ministrado de modo expositivo-construtivista, explorando a participação dos estudantes no desenvolvimento dos pontos. Utilizar-se-ão a lousa, o projetor e eventuais recursos didáticos complementares.

### Observância da frequência

A frequência do estudante nas aulas será observada através de uma chamada no início e de outra no término da aula. Para ser aprovado, o estudante precisa ter no mínimo 75% de frequência nos encontros presenciais, o equivalente a 54 horas-aula neste caso. Portanto, o estudante que tiver mais de 18 faltas estará automaticamente reprovado, sem direito a recuperação, independentemente de suas notas.

### Fórum

A página do curso no *Moodle* dispõe de um fórum, no qual o professor publicará eventuais avisos à turma e no qual os estudantes poderão postar dúvidas, organizar grupos de estudos, agendar horários de atendimento extra e debater sobre exercícios e quaisquer outros tópicos pertinentes.

## 7 Avaliação do processo ensino-aprendizagem

Todas as notas descritas são consideradas no intervalo  $[0,0;10,0]$  com precisão de uma só casa decimal, valendo sempre o arredondamento para cima.

## Composição da nota final

A nota final é dada por:

$$NF = \min\left\{10,0; \frac{LE + 0,6LE^R + AE}{2} + 0,2EF\right\}, \quad (1)$$

sendo:

- $LE$  a nota referente aos problemas resolvidos das Listas de Exercícios dentro do prazo ordinário;
- $LE^R$  a nota referente aos problemas resolvidos das Listas de Exercícios dentro do prazo extraordinário de recuperação (cf. Seção 7.1);
- $AE$  a nota referente às apresentações das Listas de Exercícios;
- $EF$  a nota referente ao Exame Final (cf. Seção 7.1).

## Listas de exercícios

Conforme o Cronograma (Seção 5), serão publicadas no Portal *URI Online Judge Academic* 18 listas de exercícios, cada uma com não mais que 5 exercícios, as quais deverão ser resolvidas individualmente no prazo de 168 horas, a contar da hora de início do encontro acadêmico em que a publicação da lista foi prevista. As resoluções dos exercícios devem ser submetidas no próprio Portal *URI Online Judge Academic*, e apenas submissões aceitas pelo Portal serão consideradas para a composição da nota. Os estudantes também poderão discutir os exercícios no fórum do *Moodle* horário de atendimento com o professor. O objetivo desta atividade é desenvolver as habilidades relacionadas a técnicas avançadas de Programação fomentar a organização de grupos de estudo. Frisa-se que haverá pouco tempo nos encontros acadêmicos para o desenvolvimento das soluções dos exercícios, necessitando que os estudantes agendem seus grupos de estudo em horários extraclasse.

A nota  $LE$  (cf. Equação 1) será dada pela razão, multiplicada por 10, entre o número de problemas resolvidos em prazo ordinário pelo estudante e o número total de problemas publicados pelo professor.

## Apresentações de exercícios

No início do encontro acadêmico em que o Cronograma (Seção 5) prevê a publicação da Lista de Exercícios  $i$ , para todo  $i \in \{2, \dots, 18\}$ , serão sorteados dois estudantes e dois problemas da Lista de Exercícios  $i - 1$ , um problema para cada estudante sorteado. O sorteio dos estudantes será feito sempre sem reposição, até que todos os estudantes tenham sido sorteados, ocasião em que todos serão repostos e o procedimento será iterado, tantas vezes quantas forem necessárias e enquanto for factível que todos os estudantes sejam sorteados um igual número de vezes. Estudantes sorteados ausentes só serão repostos ao sorteio se suas faltas forem devidamente justificadas. O sorteio dos problemas será feito apenas dentre os problemas que foram resolvidos por ao menos um estudante da turma, independentemente se o estudante designado por sorte àquele problema conseguiu resolvê-lo. Ao lhe ser designado por sorte um problema, o estudante terá 10 minutos para apresentar na lousa seus comentários sobre a solução do problema, valendo-se apenas de marcador de lousa e apagador, além da projeção do enunciado do problema. Nenhum material de consulta, de qualquer tipo, poderá ser usado. Não é necessário apresentar códigos; todavia, os comentários devem ser suficientemente claros, elegantes, formais e completos.

A cada apresentação o estudante receberá uma nota. A nota  $AE$  (cf. Equação 1) será dada pela média aritmética simples das notas atribuídas ao estudante por todas as suas apresentações.

## A penalidade por plágio ou fraude

Se for detectada a ocorrência de plágio ou fraude em qualquer instrumento de avaliação ou de recuperação, o estudante infrator terá a nota correspondente anulada, sem direito a recuperação. Se for julgado necessário, encaminhar-se-á uma denúncia à Coordenação do Curso para que medidas mais severas sejam tomadas. Adverte-se que o Portal *URI Online Judge Academic* possui uma avançada ferramenta de detecção de plágio de código.



## 7.1 Recuperação: novas oportunidades de aprendizagem e avaliação

Após expirado o prazo ordinário de 168 horas para a submissão das resoluções dos exercícios, será concedido ainda um prazo extraordinário de mais 168 horas para a resolução de exercícios pendentes, sob a aplicação de um peso de 60%. A nota  $LE^R$  (cf. Equação 1) será dada pela razão, multiplicada por 10, entre o número de problemas resolvidos pelo estudante em prazo extraordinário e o número total de problemas publicados pelo professor.

Também será aplicado no final do semestre um Exame Final, uma avaliação escrita individual sem consulta, que permitirá ao estudante acrescentar até 2 pontos à sua nota, conforme a Equação 1. As horas-aula do encontro acadêmico em que será realizado o Exame Final não serão contabilizadas na carga horária da componente curricular.

Por fim, encoraja-se que todos os estudantes submetam-se aos instrumentos avaliativos de recuperação, mesmo os com boas notas, a fim de aprimorar ainda mais seus conhecimentos.

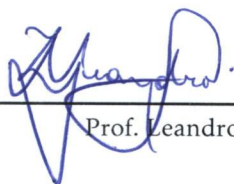
## 8 Bibliografia

### 8.1 Bibliografia básica


1. HALIM, S.; HALIM, F.  
*Competitive Programming 3: The New Lower Bound of Programming Contests*, Lulu, 2013.
2. SKIENA, S. S.; REVILLA, M.  
*Programming Challenges*, 1ª edição. Springer, 2003.
3. DASGUPTA, S.; PAPADIMITRIOU, C. H.; VAZIRANI, U.  
*Algoritmos*, McGraw Hill, 2008.
4. KLEINBERG, J.; TARDOS, E.  
*Algorithm Design*, 1ª edição. Addison Wesley, 2005.
5. SKIENA, S. S.  
*The Algorithm Design Manual*, 2ª edição. Springer, 2010.

### 8.2 Bibliografia complementar

1. BELLMAN, R.; DREYFUS, S.  
*Dynamic Programming*, 1ª edição. Princeton University Press, 2010.
2. CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C.  
*Algoritmos: Teoria e Prática*, Rio de Janeiro: Campus, 2002.
3. KNUTH, D. E.  
*The Art Of Computer Programming*, vol. 1-4 Addison-Wesley, 2011.
4. MANBER, U.  
*Introduction To Algorithms: A Creative Approach*, 1ª edição. Addison-Wesley, 1989.



Prof. Leandro M. Zatesko



MARCO AURÉLIO SPOHN  
Siape nº. 1521671  
Coord. do Curso de Ciência da Computação  
Universidade Federal da Fronteira Sul-UFFS  
Campus Chapecó-SC

Prof. Dr. Marco Aurélio Spohn  
Coordenador do Curso