



---

<b>CURSO:</b>	Engenharias	<b>SEMESTRE/ANO:</b>	01/2021
<b>DISCIPLINA:</b>	Tópicos Especiais em Programação	<b>CRÉDITOS:</b>	04
<b>CARGA HORÁRIA:</b>	60 horas		
<b>PROFESSOR:</b>	Edson Alves da Costa Júnior		

---

# PLANO DE ENSINO

## 1 Objetivos da Disciplina

A disciplina Tópicos Especiais em Programação tem como objetivo preparar os alunos do curso de Engenharia de Software da FGA para competições de programação, como a Maratona de Programação. Estes eventos ampliam o horizonte dos alunos e os estimulam a se aprofundarem nos tópicos de programação em geral. Além disso, a disciplina também constitui mais uma oportunidade para estudo e aprimoramento dos alunos em programação, tornando-os engenheiros mais preparados e capazes de atuar com competência no mercado de trabalho.

## 2 Ementa do Programa

### I. *Introdução*

- i. Programação Competitiva
- ii. Maratonas de Programação
- iii. Juízes Eletrônicos
- iv. Dicas para estudo e treinamento
- v. Como começar

### II. *Grafos*

### i. Introdução

- ii. Travessias
- iii. Árvore Geradora Mínima
- iv. Caminhos Mínimos
- v. Fluxo de Rede
- vi. Grafos Especiais

## 3 Horário das aulas e atendimento

**AULAS:** sextas, das 14:00 hrs às 18:00 hrs.

**ATENDIMENTO:** segundas e quartas, das 16:00 às 17:00 hrs.

## 4 Metodologia

Cada aula será dividida em duas etapas. A primeira etapa consistirá em encontros síncronos na plataforma Teams da Microsoft, com início às 14 horas. Nestes encontros serão debatidos os conceitos teóricos e práticos da disciplina. Caso a plataforma Teams não esteja acessível durante a aula, a reunião será feita na plataforma RNP: <https://conferenciaweb.rnp.br/webconf/edson-alves-da-costa-junior>. Estes encontros serão gravados e disponibilizados posteriormente na plataforma Youtube.

A segunda etapa será uma competição simulada com uma hora e cinquenta minutos de duração, na plataforma vJudge<sup>1</sup> ou na plataforma Codeforces<sup>2</sup>, com início às 16:00 hrs.

O Plano de Ensino, as notas e os rankings serão disponibilizados nas plataformas Moodle e Teams.

## 5 Critérios de Avaliação

A avaliação do curso se dará por meio de uma única prova, individual, cuja data está prevista no cronograma. Duas atividades extras podem alterar a menção final do aluno: o *ranking* do curso e o *contest* Menção++. A forma e os critérios destas atividades, assim como a prova, serão descritos a seguir.

### 5.1 Prova

A prova  $P$  será composta por 9 ou 10 problemas, a serem resolvidos individualmente. É permitida a consulta a matérias impressos e é vedada a consulta aos colegas ou a recursos online.

A solução proposta para um problema será corrigida de acordo com os seguintes critérios: após ser compilada de forma bem sucedida, uma série de testes unitários automatizados alimentarão o programa resultante com entradas válidas e comparará os resultados obtidos com as saídas corretas. Uma solução será considerada aceita se obtiver sucesso em todos os testes unitários.

A menção final do curso será dada em função número  $N$  de problemas cujas soluções foram aceitas, de acordo com a tabela abaixo.

$N$	Menção	Descrição
0	SR	<i>Sem rendimento</i>
1	II	<i>Inferior</i>
2 ou 3	MI	<i>Médio inferior</i>
4 ou 5	MM	<i>Médio</i>
6 ou 7	MS	<i>Médio superior</i>
8 ou mais	SS	<i>Superior</i>

<sup>1</sup><https://vjudge.net>

<sup>2</sup><http://codeforces.com>

## 5.2 Critérios de aprovação

Obterá **aprovação** no curso o aluno que cumprir as **duas** exigências abaixo:

1. Ter presença em 75% ou mais das aulas;
2. Obter menção igual ou superior a MM.

## 5.3 *Ranking* do curso

Ao longo do curso será mantido um *ranking* dos alunos do curso, em ordem decrescente de *score*. O *score*  $S$  é composto por 4 variáveis: auto-avaliação ( $A$ ), *ranking* do Codeforces ( $R$ ), listas de exercícios ( $L$ ), *contests* em sala de aula ( $C$ ):

$$S = 100A + 200L + 300R + 400C$$

Em caso de empate, serão considerados os valores de  $C$ ,  $R$ ,  $L$  e  $A$ , nesta ordem. Persistindo o empate, será utilizada a ordem lexicográfica do nome completo do estudante. Farão parte do *ranking* apenas os estudantes que satisfizerem os seguintes critérios:

1.  $R > 0$
2.  $C > 0$
3.  $L \geq 0.4$

Ao final do semestre, após a última aula que antecede a prova, posições finais do *ranking* corresponderão às menções conquistadas, de acordo com a tabela abaixo.

Posição	Menção	Descrição
[1, 4]	SS	<i>Superior</i>
[5, 11]	MS	<i>Médio superior</i>
[12, 20]	MM	<i>Médio inferior</i>

O aluno que conquistar uma menção de acordo com os critérios anteriores poderá tentar uma melhor menção na prova.

### 5.3.1 Auto-avaliação

Ao final do semestre, antes da prova, será disponibilizado um questionário para que o estudante possa fazer sua auto-avaliação. A nota  $A$  associada a este critério será computada a partir das respostas do estudante.

### 5.3.2 *Ranking* do Codeforces

A nota  $R$  do estudante será dada em função do melhor *rating*  $R_{\max}$  obtido pelo aluno na plataforma Codeforces<sup>3</sup> ao longo do semestre. O valor de  $R$  será dado por

$$R = \frac{\min(2200, \max(800, R_{\max})) - 800}{1400},$$

se o estudante participou de 3 ou mais eventos no Codeforces ao longo do semestre, ou  $R = 0$ , caso contrário.

### 5.3.3 Listas de exercícios

A cada aula será proposta uma lista de exercícios na plataforma vJudge. Estas listas terão duração de 15 dias e são individuais. Cada lista terá  $N_i$  exercícios a serem resolvidos e, após seu encerramento, as novas soluções não serão contabilizadas para o *ranking*.

É vedada a colaboração entre os estudantes da disciplina e a consulta a soluções de outras pessoas. A verificação de algum indício de fraude ou cola em qualquer um dos exercícios da lista resultará na exclusão imediata do aluno do *ranking*.

O valor de  $L$  é dado por

$$L = \frac{\sum_i AC_i}{\sum_i N_i},$$

onde  $AC_i$  é o número de exercícios com veredito AC na lista  $i$ .

### 5.3.4 *Contests* semanais

Na segunda parte de cada aula acontecerá um *contest* individual, a ser feitos na plataforma Codeforces ou na plataforma vJudge<sup>4</sup>. Cada *contest* terá  $N_i$  exercícios a serem resolvidos, com início às 16:00 hrs e término às 17:50 hrs.

É vedada a colaboração entre os estudantes e a consulta à soluções de outras pessoas ou materiais digitais. É permitida a consulta a material impresso. A verificação de algum indício de fraude ou cola em qualquer um dos *contests* resultará na exclusão imediata do aluno do *ranking* da disciplina.

O valor de  $C$  é dado por

$$C = \frac{1}{N} \sum_{i=1}^N \frac{D_i - P_i + 1}{D_i}$$

onde  $D_i$  é o número de estudantes que participaram do *contest*  $i$  e  $P_i$  é a posição final do estudante no *contest*  $i$ .

---

<sup>3</sup>[codeforces.com](https://codeforces.com)

<sup>4</sup>[vjudge.net](https://vjudge.net)

## 5.4 Contest Menção++

Após a prova  $P$ , os alunos que tiverem nota  $L \geq 0.25$  terão uma oportunidade de melhorar sua menção por meio do *contest* Menção++.

Este *contest* seguirá o mesmo formato e condições da prova  $P$ , com a diferença que a menção só poderá ser incrementada uma única vez, conforme mostra a tabela abaixo.

Menção anterior	$N$	Nova menção
II	2 ou mais	MI
MI	4 ou mais	MM
MM	6 ou mais	MS
MS	8 ou mais	SS

## 6 Cronograma

Semana	Aula	Data	Conteúdo
01	1	23/07	<i>Apresentação do curso. Introdução à Programação Competitiva.</i>
02	2	30/07	<i>Conceitos elementares. Representação de grafos</i>
03	3	06/08	<i>DFS e BFS</i>
04	4	13/08	<i>Componentes conectados. Detecção de ciclos</i>
05	5	20/08	<i>Grafos bipartidos. Pontes e pontos de articulação</i>
06	6	27/08	<i>Caminhos mínimos com fonte única</i>
07	7	03/09	<i>Caminhos mínimos entre todos os pares</i>
08	8	10/09	<i>Árvores: travessia e diâmetro</i>
09	9	17/09	<i>Árvores geradoras mínimas</i>
10	10	24/09	<i>Ordenação topológica. Programação dinâmica em DAGs</i>
11	-	01/10	<b>Semana de Extensão Universitária</b>
12	11	08/10	<i>Componentes fortemente conectados</i>
13	12	15/10	<i>Fluxo e cortes</i>
14	13	22/10	<i>Queries em árvores</i>
15	-	29/10	<b>Prova</b>
16	-	05/11	<b>Contest Menção++. Menção Final.</b>

## 7 Bibliografia

### LIVRO TEXTO

**HALIM**, Steven S. and **HALIM**, Felix. *Competitive Programming*, 1ª ed, Lulu, 2010. (*Open Access*)

**CORMEN**, Thomas H. **LEISERSON** and Charles E. and **RIVEST**, Ronald L. and **STEIN**, Clifford. *Algoritmos: Teoria e Prática*, Editora Campus, 2ª ed, 2002.

**DROZDEK**, Adam. *Estruturas de Dados e Algoritmos em C++*, Thomsom, 2001.

### LITERATURA COMPLEMENTAR

**KERNIGHAN**, Brian and **RITCHIE**, Dennis M. *The C Programming Language*, Prentice Hall, 1988.

**JOSUTTIS**, Nicolai M. *The C++ Standard Library*, Addison-Wesley, 1999.

**SOLTYS-KULINICZ**, Michael. *Introduction to the Analysis of Algorithms*, World Scientific Publishing Co, 2012. (*eBrary*)

**STEPHENS**, Rod. *Essential Algorithms: A Practical Approach to Computer Algorithms*, John Wiley & Sons, 2013. (*eBrary*)

**BALDWIN**, Douglas; **SCRAGG**, Gregg. *Algorithms and Data Structures: The Science of Computing*, Charles River Media, 2004. (*eBrary*)