



|                       |                                    |                  |          |
|-----------------------|------------------------------------|------------------|----------|
| <b>CURSO:</b>         | Engenharia de Software             |                  |          |
| <b>DISCIPLINA:</b>    | Paradigmas de Solução de Problemas |                  |          |
| <b>SEMESTRE/ANO:</b>  | 2025/1                             |                  |          |
| <b>CARGA HORÁRIA:</b> | 60 horas                           | <b>CRÉDITOS:</b> | 04       |
| <b>PROFESSOR:</b>     | Edson Alves da Costa Júnior        | <b>CÓDIGO:</b>   | FCTE0002 |

# PLANO DE ENSINO

## 1 Objetivos da Disciplina

A disciplina Paradigmas de Solução de Problemas tem como objetivo preparar os alunos do curso de Engenharia de Software da FCTE 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. Paradigmas de Solução de Problemas

- i. Busca Completa
- ii. Divisão e Conquista
- iii. Algoritmos Gulosos
- iv. Programação Dinâmica

## 3 Horário das aulas e atendimento

**AULAS:** terças e quintas, das 16:00 às 17:50 hrs.

**ATENDIMENTO:** segundas, das 12:30 às 14:30 hrs, via plataforma Teams.

## 4 Metodologia

A metodologia consiste em aulas expositivas, com o auxílio do quadro branco e projetor digital. A fim de fortalecer a aprendizagem da disciplina, as aulas serão complementadas com exercícios e atividades, presenciais e extra-classe. As comunicações do curso serão feitas exclusivamente através da plataforma SIGAA.

O curso também será focado na resolução de exercícios, envolvendo a análise e resolução de problemas oriundos de competições e de *online judges*. Ocasionalmente acontecerão contests ou na plataforma vJudge<sup>1</sup>, ou na plataforma Codeforces<sup>2</sup>, ou na plataforma AtCoder<sup>3</sup>, ou na plataforma MOJ<sup>4</sup>.

## 5 Critérios de Avaliação

A avaliação do curso se dará por meio de duas provas, individual, cujas datas estão previstas no cronograma.

### 5.1 Provas

Cada prova será composta por 10 problemas, e será aplicada em dois dias, com 5 problemas cada. É permitida a consulta a materiais impressos e é vedada a consulta aos colegas ou a recursos online. A prova terá início às 16:10 hrs, e **não serão admitidos estudantes no ambiente de provas após às 16:15 hrs.**

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.

Após a aplicação da prova, as soluções propostas pelos estudantes serão avaliadas por ferramentas de identificação de plágio, e caso duas ou mais soluções apresentem índices de similaridade que caracterizem cópia, todas elas serão anuladas, mesmo que tenha recebido o veredito “Aceito” durante a prova.

A prova será realizada, a menos de dificuldades técnicas ou de indisponibilidade de equipamentos, nas máquinas do laboratório e em ambiente Linux, por meio do Nutella Boot do professor Bruno Ribas. As soluções para os problemas devem ser escritas em C, C++ ou Python. Soluções em outras linguagens não serão aceitas.

A menção final do curso será dada pelo total  $N$  de problemas cujas soluções foram aceitas, e não anuladas, nas duas provas, de acordo com a tabela abaixo.

---

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

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

<sup>3</sup>[atcoder.jp](http://atcoder.jp)

<sup>4</sup><https://moj.naquadah.com.br/cgi-bin/index.sh>

| <i>N</i> | <b>Menção</b> | <b>Descrição</b>      |
|----------|---------------|-----------------------|
| 0        | SR            | <i>Sem rendimento</i> |
| 1 a 4    | II            | <i>Inferior</i>       |
| 5 a 9    | MI            | <i>Médio inferior</i> |
| 10 a 13  | MM            | <i>Médio</i>          |
| 14 a 17  | MS            | <i>Médio superior</i> |
| 18 a 20  | SS            | <i>Superior</i>       |

## 5.2 Listas de exercícios

A cada semana poderá ser proposta uma lista de exercícios, com exercícios relacionados com o conteúdo ministrado. A resolução das listas não modifica a menção, mas é fortemente encorajada para a fixação dos conceitos apresentados no curso.

## 5.3 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.

**IMPORTANTE:** Atestados médicos e documentos comprobatórios de justificativas de faltas dão direito à realização de atividades avaliativas que você venha a perder, mas essas ausências justificadas também são levadas em consideração como ausências efetivas para o cômputo da frequência mínima obrigatória (*Graduação UnB – Manual para estudantes*, pág. 35).

## 6 Cronograma

| <b>Semana</b> | <b>Aula</b> | <b>Data</b> | <b>Conteúdo</b>  |
|---------------|-------------|-------------|--|
| <b>01</b>     | 1           | 25/03       | <i>Apresentação do curso. Introdução à Programação Competitiva</i> |
|               | 2           | 27/03       | <i>Busca Completa: definição</i>                                   |
| <b>02</b>     | 3           | 01/04       | <i>Backtracking</i>  |
|               | -           | 03/04       | <b>Contest 1</b>   |
| <b>03</b>     | 4           | 08/04       | <i>Meet in the Middle</i>  |
|               | 5           | 10/04       | <i>Algoritmos gulosos</i>  |

| Semana | Aula | Data  | Conteúdo                                   |
|--------|------|-------|--|
| 04     | 6    | 15/04 | <i>Two pointers</i>                        |
|        | -    | 17/04 | <b>Contest 2</b>                           |
| 05     | 7    | 22/04 | <i>Divisão e Conquista: busca binária</i>  |
|        | 8    | 24/04 | <i>Divisão e Conquista: busca ternária</i> |
| 06     | 9    | 29/04 | <i>Divisão e Conquista: Mergesort</i>      |
|        | -    | 01/05 | <b>Contest 3</b>                           |
| 07     | -    | 06/05 | <b>Prova 1A</b>                            |
|        | -    | 08/05 | <b>Prova 1B</b>                            |
| 08     | 10   | 13/05 | <i>Transformada de Fourier</i>             |
|        | 11   | 15/05 | <i>Transformada Rápida de Fourier</i>      |
| 09     | 12   | 20/05 | <i>Square Root Decomposition</i>           |
|        | -    | 22/05 | <b>Contest 4</b>                           |
| 10     | 13   | 27/05 | <i>Programação Dinâmica: definição</i>     |
|        | 14   | 29/05 | <i>Prefix sum</i>                          |
| 11     | 15   | 03/06 | <i>Max range sum</i>                       |
|        | -    | 05/06 | <b>Contest 5</b>                           |
| 12     | 16   | 10/06 | <i>Problema do troco</i>                   |
|        | 17   | 12/06 | <i>Problema da mochila</i>                 |
| 13     | 18   | 17/06 | <i>Maior subsequência crescente</i>        |
|        | -    | 19/06 | <b>Ponto Facultativo: Corpus Christie</b>  |
| 14     | 19   | 24/06 | <i>Problema do Caixeiro Viajante</i>       |
|        | -    | 26/06 | <b>Contest 6</b>                           |
| 15     | 20   | 01/07 | <i>DP de dígitos</i>                       |
|        | 21   | 03/07 | <i>SOS DP</i>                              |
| 16     | 22   | 08/07 | <i>Knuth Optimization</i>                  |
|        | -    | 10/07 | <b>Contest 7</b>                           |
| 17     | -    | 15/07 | <b>Prova 2A</b>                            |
|        | -    | 17/07 | <b>Prova 2B</b>                            |
| 18     | -    | 22/07 | <b>Prova Substitutiva</b>                  |
|        | -    | 24/07 | <b>Revisão de Notas. Menções Finais</b>    |

## 7 Bibliografia

### LIVRO TEXTO

**HALIM**, Steven S. and **HALIM**, Felix. *Competitive Programming*, 4ª ed, Lulu, 2010.

**LAARKSONEN** , A. *Competitive Programmer's Handbook*, Online, 2018.

**ROUGHGARDEN** , T. *Algorithms Illuminated (Part 3): Greedy Algorithms and Dynamic Programming*, Editora LLC, 2019.

#### LITERATURA COMPLEMENTAR

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

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

**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*)