



**UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO - UNIRIO.**  
**CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA – CCET**  
**ESCOLA DE INFORMÁTICA APLICADA - EIA**

**Programa de disciplina**

CURSO: Bacharelado em Sistemas de Informação

DEPARTAMENTO: Informática Aplicada

DISCIPLINA: Análise de Algoritmos

CÓDIGO: TIN0118                      TIPO: Obrigatória

CARGA HORÁRIA: 60 horas      Nº DE CRÉDITOS: 4 créditos      PERÍODO: 5º.

PROFESSOR: Vânia Maria Félix Dias

**EMENTA:**

Crerios de análise, correção e eficiência. Complexidade de Pior Caso e Caso Médio. Divisão e Conquista; Algoritmos Gulosos; Programação Dinâmica. Modelagem em grafos. Teoria da Complexidade: Problemas de Decisão; Transformações Polinomiais; Classes P, NP, Co-NP e NP-completa.

PRÉ-REQUISITOS: TIN0168 - Estruturas de Dados II

CO-REQUISITOS: Não há

**OBJETIVOS DA DISCIPLINA:**

- Analisar e avaliar a eficiência computacional na construção de algoritmos;
- Projetar algoritmos eficientes, se possível, identificando qual método é o mais indicado em cada caso;
- Identificar e classificar a natureza de problemas computacionais.

**METODOLOGIA:**

Exposição de conteúdo: serão apresentados o conteúdo teórico das diversas técnicas de desenvolvimentos de projetos de algoritmos, com as devidas análises de correção e complexidade.

Aprendizagem colaborativa: para o entendimento dos conteúdos serão apresentados problemas cujas soluções serão propostas e discutidas pelos alunos.

Aprendizagem baseada em projeto: a fim de solucionar alguns problemas propostos, o aluno deverá desenvolver projetos de algoritmos utilizando técnicas apresentadas.

**CONTEÚDO PROGRAMÁTICO:**

1. Classes e Notação Assintótica. Complexidades de Caso Médio e Pior Caso
  - 1.1 Determinação da Complexidade de Caso Médio. Exemplos
  - 1.2 Dificuldades empíricas (determinar a distribuição, as probabilidades, etc)
2. Limite Inferior para Problemas Computacionais

- 2.1 Limite Inferior para encontrar o maior e o menor elemento de um conjunto.
- 2.2 Limite Inferior Problema de Ordenação.
3. Análises de Corretude e de Complexidade de Tempo em Algoritmos de Ordenação por comparação
  - 3.1 Algoritmos: MergeSort, InsertionSort, SelectionSort, BubbleSort, QuickSort e HeapSort
  - 3.2 Análise das complexidades dos algoritmos
  - 3.3 Algoritmos Ótimos
4. Divisão e Conquista: Estratégia, definições e aplicabilidade. Recursão e Recorrência. Técnicas de solução de Equações de Recorrência. Teorema Mestre
  - 4.1 Algoritmos: Mediana; Multiplicação de Matrizes; Multiplicação de Inteiros Grandes; Menor Distância entre Pontos
  - 4.2 Análise da corretude e complexidade dos algoritmos
5. Algoritmos Gulosos: Estratégia, definições e aplicabilidade
  - 5.1 Algoritmos: Problema da Árvore Geradora Mínima; Problema da Mochila Fracionário; Códigos de Huffman
  - 5.2 Análise da corretude e complexidade dos algoritmos
6. Programação Dinâmica: Estratégia, definições e aplicabilidade
  - 6.1 Algoritmos: Problema da Subsequência Crescente Máxima; Problema da Subsequência Comum mais Longa; Problema da Multiplicação de Cadeias de Matrizes
  - 6.2 Análise da corretude e complexidade dos algoritmos
7. Modelagem de Problemas em Grafos
  - 7.1 Problemas de caminhos mais curtos em grafos
    - Algoritmo de Dijkstra
    - Algoritmo de Bellman-Ford
    - Algoritmo de Floyd-Warshall
    - Análise da corretude e complexidade dos algoritmos
  - 7.2 Problema do fluxo máximo em grafos:
    - Algoritmo de Ford & Fulkerson
    - Análise da corretude e complexidade
  - 7.3 Outros problemas combinatórios modelados através de grafo
8. Classes de Problemas Computacionais: Problemas de Decisão, Problemas de Localização e Problemas de Otimização
  - 8.1 A classe P. As classes NP e Co-NP
  - 8.2 A classe NP-Completo
  - 8.3 Transformações Polinomiais
  - 8.4 Demonstrações de alguns Problemas NP-Completo
    - Clique;
    - Cobertura de Vértices;
    - Conjunto Independente;
    - Ciclo Hamiltoniano.
  - 8.5 Algoritmos Pseudo-Polinomiais.

## CRONOGRAMA:

Semana 1: Classes e Notação Assintótica. Complexidades de Caso Médio e Pior Caso Limite Inferior para Problemas Computacionais

Semana 2: Análises de Corretude e de Complexidade de Tempo em Algoritmos de Ordenação por comparação. Limite Inferior do problema.

Semana 3: Divisão e Conquista: Estratégia, definições e aplicabilidade. Recursão e Recorrência. Técnicas de solução de Equações de Recorrência. Teorema Mestre

Semana 4: Algoritmos: Mediana; Multiplicação de Matrizes; Multiplicação de Inteiros Grandes; Menor Distância entre Pontos

Semana 5: Análise da corretude e complexidade dos algoritmos apresentados na semana (4)

Semana 6: Algoritmos Gulosos: Estratégia, definições e aplicabilidade

Semana 7: Algoritmos: Problema da Árvore Geradora Mínima; Problema da Mochila Fracionário; Códigos de Huffman

Semana 8: Análise da corretude e complexidade dos algoritmos apresentados na semana (7)

Semana 9: Programação Dinâmica: Estratégia, definições e aplicabilidade

Semana 10: Algoritmos: Problema da Subsequência Crescente Máxima; Problema da Subsequência Comum mais Longa; Problema da Multiplicação de Cadeias de Matrizes Análise da corretude e complexidade dos algoritmos

Semana 11: Modelagem de Problemas em Grafos; Problemas de caminhos mais curtos em grafos: Algoritmo de Dijkstra, Algoritmo de Bellman-Ford, Algoritmo de Floyd-Warshall.

Semana 12: Análise da corretude e complexidade dos algoritmos apresentados na semana (11)

Semana 13: . Classes de Problemas Computacionais: Problemas de Decisão, Problemas de Localização e Problemas de Otimização. As classes P, NP e Co-NP; A classe NP-Completo

Semana 14: Transformações Polinomiais; Demonstrações de alguns Problemas NP-Completos: Clique; Cobertura de Vértices; Conjunto Independente; Ciclo Hamiltoniano. Algoritmos Pseudo-Polinomiais.

Semana 15: Avaliação Final.

#### AVALIAÇÃO:

Avaliação contínua: ao longo da disciplina, o estudante desenvolverá aproximadamente 5 projetos de algoritmos - um para cada conjunto de problemas propostos - constando as respectivas análises de correção e complexidade, sendo cada um dos problemas correspondente a uma técnica específica de projeto de algoritmos estudada.

Além dos projetos, o aluno deverá resolver listas de exercícios.

#### FERRAMENTAS DIGITAIS UTILIZADAS:

- Google Classroom
- Meet

#### BIBLIOGRAFIA BÁSICA

- Jon Kleinberg, Éva Tardos, *Algorithm Design*, Addison-Wesley, 2005. (online)
- S. Dasgupta, C.H. Papadimitriou, U.V. Vazirani, *Algorithms*, McGraw-Hill, 2006.

#### BIBLIOGRAFIA COMPLEMENTAR

- Szwarcfiter, J.L. Teoria Computacional de Grafos. Rio de Janeiro, Elsevier, 2018.
- Cormen, T.H.; Leiserson, C.E.; & Rivest, R.L.; Stein, C. *Introduction to Algorithms*, 3rd ed.. The MIT Press, 2009.
- T.H. Cormen, *Algorithms Demystified*, MIT Press, 2012.
- Sedgewick, Robert; Wayne, Kevin. *Algorithms*. 4ª ed. Addison-Wesley Professional, 2011.
- Knuth, D.E. *The Art of Computer Programming - Vols I, II e III*. 2nd Edition. Addison Wesley, 1973.
- Szwarcfiter, J.L. & Markenzon, L. *Estruturas de Dados e seus Algoritmos*. 3ª ed. Rio de Janeiro, LTC, 2010.
- Apostila: [https://www.ime.usp.br/~pf/analise\\_de\\_algoritmos/](https://www.ime.usp.br/~pf/analise_de_algoritmos/)

Assinatura do professor

