



UNIVERSIDADE FEDERAL DE OURO PRETO
DEPARTAMENTO DE COMPUTAÇÃO



PLANO DE ENSINO

Nome do Componente Curricular em português: Estruturas de Dados I		Código: BCC202
Nome do Componente Curricular em inglês: Data Structures I		
Nome e sigla do departamento: Departamento de Computação (DECOM)		Unidade acadêmica: ICEB
Nome do docente: Pedro Henrique Lopes Silva		
Carga horária semestral: 90 horas	Carga horária semanal teórica: 4 horas/aula	Carga horária semanal prática: 2 horas/aula
Data de aprovação na assembleia departamental: 18/10/2024		
Ementa: Recursividade; conceitos básicos de análise assintótica de algoritmos; tipos abstratos de dados; estruturas de dados: listas, pilhas, filas de prioridade e árvores binárias; algoritmos de ordenação por comparação de chaves: seleção, inserção, bolha, shellsort, quicksort, mergesort, heapsort; algoritmos de ordenação em tempo linear: counting sort, radix sort e bucket sort; e algoritmos de pesquisa: simples, binária, árvores binárias de busca e hashing.		
Conteúdo Programático: <ul style="list-style-type: none">• Revisão de alocação dinâmica de memória• Recursividade• Noções de análise de complexidade de algoritmos:• Conceitos• Medidas de avaliação: tempo e espaço• Análise assintótica: notação O, Omega e Theta• Hierarquia de funções e classes de problemas• Tipos de dados abstratos• Estruturas de Dados• Listas• Pilhas• Filas• Filas de prioridade• Árvores• Algoritmos• Métodos de ordenação por comparação: Selection Sort, Insertion Sort, Bubblesort, Shellsort, Quicksort, Heapsort e Mergesort• Métodos de ordenação em tempo linear: Counting Sort, Radix Sort e Bucket Sort• Métodos de pesquisa: Simples, Binária, Árvores Binárias e AVL e Hashing		

Objetivos:

O aluno deverá conhecer conceitos associados a tipos abstratos de dados e métodos de pesquisa e ordenação de interesse teórico e prático.

Deverá também adquirir a capacidade de utilizar esses recursos pra desenvolvimento de programas, utilizando conceitos de modularização e abstração de dados.

Deverá ainda ser capaz de comparar estratégias de implementação do ponto de vista da complexidade dos algoritmos envolvidos, usando a notação O.

Metodologia:

Aulas presenciais: As atividades envolvem:

- Preparação de aulas e outros tipos de materiais complementares.
- Leituras recomendadas de textos técnicos, apostilas e livros de autores especialistas com a finalidade de proporcionar ao discente a oportunidade de consulta e desenvolvimento de sua capacidade de análise, síntese e crítica de uma bibliografia específica.
- Aulas práticas (P) com a implementação dos métodos estudados. O professor proverá auxílio durante o horário da aula prática. Exercícios resolvidos deverão ser enviados em formato de código-fonte pelo o ambiente virtual de aprendizagem adotado.
- Trabalhos práticos abrangendo todo o conteúdo da disciplina. Os alunos deverão entregar o trabalho pelo ambiente virtual de aprendizagem adotado e apresentá-lo para o(s) professor(es) em um horário específico se assim for solicitado.

Recursos utilizados: Para o desenvolvimento da aprendizagem serão adotados, concomitantemente e em todos os tópicos da disciplina, os seguintes recursos de apoio didático-pedagógico:

- Ferramentas do ambiente virtual de aprendizagem adotado (Moodle e corretor automático).
- Computador desktop ou notebook.
- Livros, Apostila, textos científicos e documentos legais em formato impresso ou digital (disponíveis gratuitamente online) para estudos extraclasse.
- As implementações das aulas práticas e trabalhos práticos necessitam de um ambiente de programação e compilação. O aluno pode optar por instalar as ferramentas necessárias (compilador e editor de texto), que são gratuitas, ou utilizar um ambiente web que não requer nenhuma instalação.

Atividades avaliativas:

- Três Provas Teóricas de 10 pontos cada mas com pesos diferentes.
- Três Trabalhos Práticos (TPs) de 10 pontos.
- Avaliações Contínuas (AC) de 10 pontos.

Média Final = $(2 * Prova1 + 2 * Prova2 + 2,5 * Prova3 + 1 * TP I + 1 * TP II + 1 * TP III + 0,5 * AC)/10$

- Provas Teóricas: os alunos receberão uma prova com questões referentes aos conteúdos estudados e terão o horário da aula para fazer e entregar a prova. As provas serão aplicadas nos horários cadastrados das disciplinas/turmas.
- Avaliações Contínuas: aulas práticas referentes aos conteúdos estudados distribuídos ao longo do semestre e atividades teóricas ao longo do período.
- Frequência: a computação da frequência será mediante a presença em aula.

- Exame Especial: O Exame Especial será uma prova única e individual. Cada aluno receberá uma prova com questões referentes aos conteúdos estudados e terão o tempo da aula para resolver e entregar.

Resolução CEPE 2880 de 05/2006: É assegurado a todo aluno regularmente matriculado com frequência mínima de **setenta e cinco por cento** e média inferior a seis, o direito de ser avaliado por Exame Especial.

Cronograma:

Atividades propostas por semana:

Semana 1	-	12/11/24	-	Apresentação do Curso
	-	14/11/24	-	Princípios da Programação em C
Semana 2	-	19/11/24	-	Alocação Dinâmica
	-	21/11/24	-	Tipos Abstratos de Dados (TADs)
Semana 3	-	26/11/24	-	Recursividade
	-	28/11/24	-	Análise Assintótica de Algoritmos I
Semana 4	-	03/12/24	-	Análise Assintótica de Algoritmos II
	-	05/12/24	-	Análise Assintótica de Algoritmos III
Semana 5	-	10/12/24	-	Recursividade e Análise Assintótica
	-	12/12/24	-	Aula de Dúvidas Prova I
Semana 6	-	17/12/24	-	Prova I e Entrega do TP I
	-	19/12/24	-	Listas
Semana 7	-	21/01/25	-	Listas Encadeadas
	-	23/01/25	-	Pilhas
Semana 8	-	28/01/25	-	Filas
	-	30/01/25	-	Métodos de Ordenação
Semana 9	-	04/02/25	-	MergeSort
	-	06/02/25	-	QuickSort
Semana 10	-	11/02/25	-	ShellSort
	-	13/02/25	-	Aula de Dúvidas Prova II
Semana 11	-	18/02/25	-	Prova II e Entrega do TP II
	-	20/02/25	-	Fila de Prioridade
Semana 12	-	25/02/25	-	HeapSort
	-	27/02/25	-	Ordenação em Tempo Linear
Semana 13	-	04/03/25	-	Recesso Acadêmico
	-	06/03/25	-	Pesquisa Sequencial e Pesquisa Binária
Semana 14	-	11/03/25	-	Árvores Binárias de Pesquisa
	-	13/03/25	-	Árvores Binárias de Pesquisa II
Semana 15	-	18/03/25	-	Árvores AVL
	-	20/03/25	-	Tabelas Hash por Lista Encadeada
Semana 16	-	25/03/25	-	Tabelas Hash Endereçamento Aberto
	-	27/03/25	-	Aula de Dúvidas Prova III
Semana 17	-	01/04/25	-	Prova III e Entrega do TP III
	-	03/04/25	-	Correção da Prova
Semana 18	-	08/04/25	-	Exame Especial

Bibliografia Básica:

- ZIVIANI, Nivio. Projeto de algoritmos: com implementações em Pascal e C. 3. ed. rev. e ampl. São Paulo: Cengage Learning, c2011. xx, 639 p. ISBN 9788522110506.
- CELES, Waldemar; CERQUEIRA, Renato; RANGEL, Jose Lucas. Introdução a Estruturas de Dados: com técnicas de programação em C.. Rio de Janeiro: Elsevier 2004. 293 p.
- CORMEN, Thomas H. Algoritmos: teoria e prática. Rio de Janeiro: Campus, 2002. xvii, 916 p.

Bibliografia Complementar:

- KLEINBERG, Jon; TARDOS, Eva. Algorithm design. Boston: Pearson/Addison-Wesley, c2006. xxiii, 838 p.
- KNUTH, Donald Ervin. The art of computer programming. Upper Saddle River: Addison Wesley, c2005. v.
- GOODRICH, Michael T.; TAMASSIA, Roberto; COPSTEIN, Bernardo. Projeto de algoritmos: fundamentos, análise e exemplos da internet. Porto Alegre: Bokman, 2004. 696 p.
- DROZDEK, Adam. Estrutura de dados e algoritmos em C++. São Paulo: Cengage Learning, c2002. xviii, 574 p.
- TENENBAUM, Aaron M; LANGSAM, Yedidiah; AUGENSTEIN, Moshe. Estruturas de dados usando C. São Paulo: Makron Books, 1995. 884 p.