



PLANO DE ENSINO SEMESTRAL

DISCIPLINA: Estruturas de Dados

CURSO: Ciência da Computação e Engenharia de Software

SEMESTRE: 2º

TURNO: Matutino

DOCENTE RESPONSÁVEL: Jean Nunes Araujo

E-MAIL: jean.araujo@idp.edu.br

CARGA HORÁRIA: 80 h/a

ANO E SEMESTRE DE REFERÊNCIA: 2024/2

HORÁRIO DAS AULAS: Quarta-feira das 10h00 às 11h40 e Sexta-feira das 08h00 às 09h40

LOCAL: Quarta-feira: Laboratório Dell (Térreo); Sexta-feira: Maker Space (Térreo).

EMENTA

Ponteiros. Alocação estática e dinâmica de memória. Tipos abstratos de dados: conceitos, operações, representações e manipulação. Estruturas de Dados: vetores, listas lineares, pilhas, filas, árvores e grafos. Algoritmos para construção, consulta e manipulação de estruturas de dados. Aplicações de estruturas de dados. Desenvolvimento de algoritmos e resolução de problemas computacionais.

OBJETIVOS

Objetivo Geral:

- Introdução às várias estruturas da informação, buscando habilitar os estudantes no uso desses recursos no desenvolvimento de tecnologias, protocolos e aplicações.

Objetivos Específicos:

- Fornecer uma base sólida sobre tipos de dados e as operações associadas aos tipos;
- Definir os objetos que constituem o dado e as operações aplicáveis fazendo uso das estruturas de dados estáticas básicas: lista, pilhas, filas e árvores;
- Demonstrar os principais algoritmos para construção, consulta e manipulação de estrutura de dados que envolvam ordenação e pesquisa;
- Orientar sobre quando e como aplicar corretamente as estruturas de dados para resolver problemas computacionais.

METODOLOGIA DE ENSINO

O objetivo das metodologias ativas é estabelecer uma parceria com o discente na busca pelo conhecimento. Busca-se incentivar o protagonismo do estudante no processo de aprendizagem através de estímulos que visam despertar a sua autonomia intelectual (Anastasiou, 2004). Para tanto, as atividades desta disciplina visam promover o uso de diversas habilidades de pensamento, tais como interpretar, questionar, analisar, sintetizar e comparar.

A metodologia de ensino base da disciplina é a aprendizagem baseada em problemas usando uma abordagem de programação em pares (Oakley, 2004). Nessa abordagem, os estudantes são organizados em pares e devem percorrer as fases da metodologia da problematização considerando as estruturas de dados abordadas.

O conteúdo da disciplina será apresentado usando uma abordagem teórico/prática, de forma que, em cada aula, haverá um momento teórico, no qual serão apresentados os recursos e estruturas lógicas a serem utilizadas e, posteriormente, um momento onde serão desenvolvidas atividades práticas em laboratório para que o acadêmico consiga compreender a aplicação dos conceitos na solução de problemas com uso do computador.

RECURSOS DIDÁTICOS

Quadro branco, pinceis (azul, preto e vermelho), apagador, *datashow*, *notebook*, internet, livros, Laboratório, Maker Space, Ambiente Virtual de Aprendizagem.

CONTEÚDO PROGRAMÁTICO

1. Ponteiros e alocação de memória.
2. Recursividade.
3. Algoritmos e Estrutura de Dados: conceitos, tipos de dados e aplicações.
4. Estruturas de Dados: Listas encadeadas, pilhas, filas, dicionários e árvores (árvores binárias, árvores de busca).
5. Algoritmos de ordenação.
6. Algoritmos de pesquisa.

AVALIAÇÃO

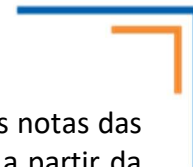
A nota final (NF) será calculada a partir da Eq. 01, que corresponde à média aritmética simples das notas das avaliações 1 e 2 (A_1 e A_2 , respectivamente).

$$NF = \frac{(A_1 + A_2)}{2}, 0 \leq NF \leq 10 \text{ (Equação 01)}$$

Nesta disciplina, os instrumentos avaliativos que irão compor as notas AV_1 e AV_2 consistem em: Provas, Listas de Exercícios e Trabalhos Práticos, sendo que cada um dos instrumentos receberá nota de 0 (zero) a 10 (dez).

A nota AV_1 será composta por seis (5) atividades práticas e uma (1) prova. Para compor as notas das atividades práticas, a atividade de menor nota será desconsiderada. A nota será calculada a partir da fórmula expressa pela Eq. 02:

$$AV_1 = \left(\left(\frac{AP_{1^{am}} + AP_{2^{am}} + AP_{3^{am}} + AP_{4^{am}}}{4} \right) * 0,4 \right) + (NP * 0,6) \text{ (Equação 02)}$$



A nota AV_2 será composta por cinco (5) atividades práticas e uma (1) prova. Para compor as notas das atividades práticas, a atividade de menor nota será desconsiderada. A nota será calculada a partir da fórmula expressa pela Equação 03:

$$AV_1 = \left(\left(\frac{AP_{1^{a}mn} + AP_{2^{a}mn} + AP_{3^{a}mn} + AP_{4^{a}mn}}{4} \right) * 0,4 \right) + (NP * 0,6) \text{ (Equação 03)}$$

Tanto na Eq. 02 quanto na Eq. 03, o subscrito “1^amn” significa “primeira maior nota”. Os demais subscritos seguem a ordem representada pelo número.

Ainda em relação às atividades práticas, cada semana de atraso na sua entrega implicará em penalidade de 10%, de forma cumulativa, na nota da atividade, de forma que poderá chegar a 100% de penalidade quando a entrega ocorrer com 10 ou mais semanas de atraso.

Além disso, será verificada a presença dos alunos em todas as aulas e atividades programadas da disciplina e, de acordo com o Regimento Geral da Instituição, será exigida frequência mínima de 75% para aprovação.

Portanto, será considerado aprovado o estudante que obtiver NF maior ou igual a 6 (seis) e frequência mínima de 75%.

Observações:

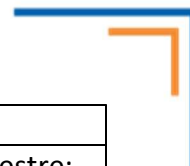
- O aluno que deixar de realizar uma avaliação somativa do tipo prova (NP_n) na data especificada pelo docente, tem direito a prova substitutiva mediante requerimento dentro do prazo indicado no Calendário Acadêmico. Dessa forma, o estudante que, por motivo de doença, deixou de fazer um teste e um estudo de caso, por exemplo, não terá direito à prova substitutiva para a atividade prática e a atividade prática integradora em questão.
- É assegurado ao discente o direito de exigir revisão dos resultados obtidos nas avaliações 1 e 2, ou seja, AV_1 e AV_2 , assim como daqueles alcançados nas provas substitutivas, desde que respeite o prazo para solicitação especificado no Calendário Acadêmico e atenda as normas aprovadas pelo Consaepe. Isso significa que, ao requerer revisão de nota da Avaliação 2 (AV_2), por exemplo, o docente irá reavaliar todas as atividades propostas para compor essa nota.

FREQUÊNCIA

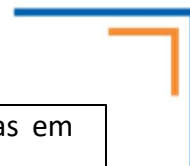
Conforme diretrizes do Ministério da Educação e Regulamento da Graduação do IDP:

- O/a discente deverá estar presente em 75% das aulas para aprovação na disciplina;
- Cada aula de 1h40 computará duas presenças ou ausências (duas aulas de 50min);
- O/a discente terá até 15min de tolerância após o início da aula para entrada em sala, após esse período não será permitida a entrada e o/a discente receberá ausência nesta aula;
- O/a discente poderá acompanhar sua frequência no Portal do Aluno e no App do Aluno.

CRONOGRAMA DE ATIVIDADES



AULA	DATA	CONTEÚDO
1	07/agosto	Apresentação da disciplina e do plano de ensino do semestre; Semana de Integração
2	09/agosto	Revisão geral de lógica de programação: fluxo e estruturas.
3	14/agosto	Ponteiros e manipulação de arquivos: formas de implementação na linguagem C.
4	16/agosto	Ponteiros e manipulação de arquivos: formas de implementação na linguagem C.
5	21/agosto	Ponteiros e manipulação de arquivos: formas de implementação na linguagem C (Exercícios e atividades práticas).
6	23/agosto	Alocação dinâmica e Alocação estática.
7	28/agosto	Recursividade: conhecer, escrever e exercitar funções recursivas.
8	30/agosto	Algoritmos recursivos: Apresentar e detalhar alguns algoritmos recursivos.
9	04/setembro	Listas encadeadas: formas de implementação de lista encadeada simples com as operações: inserção, modificação, remoção e impressão em linguagem C.
10	06/setembro	Listas encadeadas: formas de implementação de lista encadeada simples com as operações: inserção, modificação, remoção e impressão em linguagem C (Exercícios e atividades práticas).
11	11/setembro	Listas duplamente encadeadas: formas de implementação de listas duplamente encadeadas em linguagem C.
12	13/setembro	Listas duplamente encadeadas: formas de implementação de listas duplamente encadeadas em linguagem C.
13	18/setembro	Listas duplamente encadeadas: formas de implementação de listas duplamente encadeadas em linguagem C (Exercícios e atividades práticas).
14	20/setembro	Listas circulares: formas de implementação de listas circulares em linguagem C.
15	25/setembro	Listas circulares: formas de implementação de listas circulares em linguagem C (Exercícios e atividades práticas).
16	27/setembro	Pilhas: formas de implementação e suas generalizações em linguagem C.
17	02/setembro	Pilhas: formas de implementação e suas generalizações em linguagem C.
18	04/outubro	Pilhas: formas de implementação e suas generalizações em linguagem C (Exercícios e atividades práticas).
19	09/outubro	Realização da 1ª avaliação (A1)
20	11/outubro	Filas estáticas: formas de implementação de filas estáticas em linguagem C.
21	16/outubro	Filas estáticas: formas de implementação de filas estáticas em linguagem C.
22	18/outubro	Filas estáticas: formas de implementação de filas estáticas em linguagem C (Exercícios e atividades práticas).



23	23/outubro	Filas dinâmicas: formas de implementação de filas dinâmicas em linguagem C.
24	25/outubro	Filas dinâmicas: formas de implementação de filas dinâmicas em linguagem C.
25	30/outubro	Filas dinâmicas: formas de implementação de filas dinâmicas em linguagem C (Exercícios e atividades práticas).
26	01/novembro	Árvores: formas de implementação de árvores binárias e árvores de busca.
27	06/novembro	Árvores: formas de implementação de árvores binárias e árvores de busca.
28	08/novembro	Árvores: formas de implementação de árvores binárias e árvores de busca (Exercícios e atividades práticas).
29	13/novembro	Algoritmos de ordenação.
30	15/novembro	Algoritmos de ordenação.
31	20/novembro	FERIADO
32	22/novembro	Algoritmos de ordenação (Exercícios e atividades práticas).
33	27/novembro	Algoritmos de busca.
34	29/novembro	Algoritmos de busca.
35	04/dezembro	Algoritmos de busca (Exercícios e atividades práticas).
36	06/dezembro	Realização da 2ª avaliação (A2)
37		
38		
39		
40		

BIBLIOGRAFIA

BIBLIOGRAFIA BÁSICA

DROZDEK, Adam. Estrutura de dados e algoritmos em C++. 2. ed. São Paulo: Cengage Learning, 2018. E-book. (1 recurso online). ISBN 9788522126651. Disponível em:

<https://integrada.minhabiblioteca.com.br/books/9788522126651>.

CELES, Waldemar; CERQUEIRA, Renato; RANGEL, José Lucas. Introdução a estruturas de dados com técnicas de programação em C. 2. ed. Rio de Janeiro: GEN LTC, 2016. E-book. (1 recurso online). (SBC (Sociedade Brasileira de Computação)). ISBN 9788595156654. Disponível em:

<https://integrada.minhabiblioteca.com.br/books/9788595156654>.



PINTO, Rafael Albuquerque et al. Estrutura de dados. Porto Alegre: SAGAH, 2020. E-book. (1 recurso online). ISBN 9786581492953. Disponível em:
<https://integrada.minhabiblioteca.com.br/books/9786581492953>.

BIBLIOGRAFIA COMPLEMENTAR

BACKES, André Ricardo. Algoritmos e estruturas de dados em linguagem C. Rio de Janeiro: LTC, 2023. E-book. (1 recurso online). ISBN 9788521638315. Disponível em:
<https://integrada.minhabiblioteca.com.br/books/9788521638315>.

CURY, Thiago Espíndola et al. Estrutura de dados. Porto Alegre: SAGAH, 2018. E-book. (1 recurso online). ISBN 9788595024328. Disponível em:
<https://integrada.minhabiblioteca.com.br/books/9788595024328>.

SZWARCFITER, Jayme Luiz; MARKENSON, Lilian. Estruturas de dados e seus algoritmos. 3. ed. Rio de Janeiro: LTC, 2010. E-book. (1 recurso online). ISBN 978-85-216-2995-5. Disponível em:
<https://integrada.minhabiblioteca.com.br/books/978-85-216-2995-5>.

AGUILAR, Luis Joyanes. Fundamentos de programação: algoritmos, estruturas de dados e objetos. 3. ed. Porto Alegre: AMGH, 2008. E-book. (1 recurso online). ISBN 9788580550146. Disponível em:
<https://integrada.minhabiblioteca.com.br/books/9788580550146>.

AGUILAR, Luis Joyanes. Programação em c ++: algoritmos, estruturas de dados e objetos. 2. ed. Porto Alegre: AMGH, 2008. E-book. (1 recurso online). ISBN 9788580550269. Disponível em:
<https://integrada.minhabiblioteca.com.br/books/9788580550269>.

Brasília-DF, 05 de agosto de 2024.

Prof. Jean Nunes Araujo