

<b>Caracterização da disciplina</b>										
Código da disciplina:	<b>MCTA001-17</b>		Nome da disciplina:		<b>Algoritmos e Estruturas de Dados I</b>					
Créditos (T-P-I):	<b>(2-2-4)</b>		Carga horária:	<b>48 horas</b>	Aula prática:	<b>S</b>	Câmpus:	<b>SA</b>		
Código da turma:	<b>DA1MCTA001-17SA DA2MCTA001-17SA</b>		Turma:	<b>A1, A2</b>	Turno:	<b>Matutino</b>	Quadrimestre:	<b>1</b>	Ano:	<b>2020</b>
Docente(s) responsável(is):			<b>A1: MARCIO K. OIKAWA (T) / MARCIO K. OIKAWA (P) A2: MARCIO K. OIKAWA (T) / FERNANDO T. FERREIRA (P)</b>							

<b>Alocação da turma</b>						
	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
8:00 - 9:00					Teoria	
9:00 - 10:00					Teoria	
10:00 - 11:00			Prática			
11:00 - 12:00			Prática			
12:00 - 13:00						
13:00 - 14:00						
14:00 - 15:00						
15:00 - 16:00						
16:00 - 17:00						
17:00 - 18:00						
18:00 - 19:00						
19:00 - 20:00						
20:00 - 21:00						
21:00 - 22:00						
22:00 - 23:00						

<b>Planejamento da disciplina</b>			
<b>Objetivos gerais</b>			
Apresentar as diversas estruturas de dados fundamentais, como estruturas lineares (listas encadeadas, pilhas, filas, etc.), estruturas não-lineares (árvores), os algoritmos básicos para a sua manipulação, assim como as suas aplicações. Introduzir noções básicas de complexidade de algoritmos e técnicas básicas para comparação dos tempos de execução dos algoritmos estudados. Apresentar a importância da escolha da estrutura de dados e algoritmos adequados para a resolução de problemas de maneira eficiente.			
<b>Objetivos específicos</b>			
<b>Ementa</b>			
Introdução à linguagem C. Noções básicas sobre análise da complexidade de tempo de algoritmos. Estruturas de dados básicas: listas encadeadas, pilhas, filas e árvores. Busca e ordenação. Árvores de busca.			
<b>Conteúdo programático</b>			
Aula	Conteúdo	Estratégias didáticas	Avaliação
12/02 (P)	Apresentação da disciplina. Apresentação do ambiente de programação. Exercícios simples utilizando a linguagem de programação C.	Exercícios em laboratório.	Discussão e/ou exercícios
14/02 (T)	Introdução. Vetores. Ponteiros em C. Estruturas em C. Recursividade.	Aula expositiva e discussão.	Discussão e/ou exercícios
19/02 (P)	Vetores. Operações básicas com ponteiros.	Exercícios em laboratório.	Discussão e/ou exercícios
21/02 (T)	Estruturas lineares estáticas. Vetores. Conceitos de pilha e fila.	Aula expositiva e discussão.	Atividades em laboratório
26/02 (P)	Recesso	Recesso.	Recesso.
28/02 (T)	Busca em estruturas lineares. Custos de algoritmos.	Aula expositiva e discussão.	Discussão e/ou exercícios
04/03 (P)	Estruturas lineares. Algoritmos de inserção, remoção e busca.	Exercícios em laboratório.	Discussão e/ou exercícios
06/03 (T)	Estruturas lineares dinâmicas. Alocação dinâmica de memória. Implementação de pilhas e filas.	Aula expositiva e discussão.	Atividades em laboratório
11/03 (P)	Estruturas lineares dinâmicas.	Exercícios em laboratório.	Discussão e/ou exercícios
13/03 (T)	Árvores binárias. Algoritmos básicos de busca.	Aula expositiva e discussão.	Discussão e/ou exercícios
*22/04 (P)	Revisão sobre árvores. Árvores binárias. Altura de árvores binárias. Algoritmos de busca em árvores binárias.	Exercícios para entrega online.	Discussão e/ou exercícios
*24/04 (T)	Árvores binárias. Algoritmos básicos de inserção. Algoritmos básicos de remoção.	Vídeos explicativos e exercícios. Dúvidas em lista de discussão.	Discussão e/ou exercícios.
*29/04 (P)	Exercícios sobre árvores binárias.	Exercícios para entrega online.	Discussão e/ou exercícios
*01/05 (T)	Árvores balanceadas. Árvores AVL. Operações básicas: busca. Rotações.	Vídeos explicativos e exercícios. Dúvidas em lista de discussão.	Discussão e/ou exercícios
*06/05 (P)	Exercícios sobre árvores AVL.	Exercícios para entrega online.	Discussão e/ou exercícios

*08/05 (T)	Árvores AVL – operações básicas: inserção e remoção.	Vídeos explicativos e exercícios. Dúvidas em lista de discussão.	Atividades em laboratório
*13/05 (P)	Ordenação de vetores. Algoritmos básicos: bubblesort, selection sort.	Exercícios para entrega online.	Discussão e/ou exercícios
*15/05 (T)	Ordenação de vetores. Algoritmos básicos: insertion sort. Tempo de execução de algoritmos.	Vídeos explicativos e exercícios. Dúvidas em lista de discussão.	Discussão e/ou exercícios
*20/05 (P)	Ordenação de vetores. Algoritmos ótimos: mergesort, heap sort.	Exercícios para entrega online.	Discussão e/ou exercícios
*22/05 (T)	Ordenação de vetores. Algoritmos ótimos: quicksort.	Vídeos explicativos e exercícios. Dúvidas em lista de discussão.	Discussão e/ou exercícios.
*27/05 (P)	Revisão	Exercícios para entrega online.	Exercícios em laboratório.
*29/05 (T)	Prova	Avaliação individual	Avaliação online
*03/06 (P)	Prova sub.	Exercícios para entrega online.	Avaliação online
*05/06(T)	Prova REC.		Avaliação online
	*Alteração implementada no ECE.		

**Descrição dos instrumentos e critérios de avaliação qualitativa**

**Ferramentas Necessárias:** Computadores com um compilador para a linguagem de programação C, editores de texto, software para visualização de vídeos em formato mp4, leitor de arquivos em formato PDF.

**Critérios de Avaliação:** Os critérios de avaliação foram modificados para atender aos princípios propostos na Resolução ConsEPE no. 239/2020, de 06 de abril de 2020.

Os alunos são avaliados por meio de instrumentos principais: Exercícios práticos de fixação online e uma prova online. Os exercícios práticos consistem no desenvolvimento de códigos de programação gerados por computador e sua posterior avaliação em ambiente Moodle. O cálculo do conceito final será realizado considerando os seguintes critérios:

Exercícios de Fixação (EF) – 60%  
Prova online (P) – 40%

Obs.: Caso não seja possível aplicar a prova (P) na modalidade online por algum fator técnico, a mesma será deslocada para o período pós-retorno de aulas presenciais. A turma será informada oportunamente a respeito dos detalhes.

**CÁLCULO DO CONCEITO FINAL:**

Para fins de cálculo, esta disciplina usará valores numéricos para melhor refletir a influência dos pesos de cada uma das avaliações. Sendo assim, cada uma das avaliações terá atribuída uma nota numérica real entre 0,0 e 10,0. Para cada uma das avaliações, será usada a seguinte tabela de conversão:

Pontuação	Conceito
9,0 a 10,0	A
7,5 a 8,9	B
6,0 a 7,4	C
5,0 a 5,9	D
0,0 a 4,9	F

O conceito final (CF) considerará a seguinte fórmula de cálculo baseada na nota numérica obtida em cada avaliação:

$$CF = 0,4 \times P + 0,6 \times EF$$

O valor numérico CF será convertido em conceito usando também a tabela de conversão acima.

#### **AValiação Substitutiva (SUB):**

Em cumprimento à Resolução ConsEPE no. 227/2018, os alunos que não puderem entregar a prova nos casos previstos na resolução citada terão direito a uma única avaliação substitutiva. Para tal, está previsto no cronograma uma data específica, no final do quadrimestre, para realização da avaliação. O conteúdo da avaliação substitutiva é o conteúdo integral do quadrimestre e o conceito (ou nota) obtido nessa avaliação substituirá o conceito "F" atribuído à atividade na qual o(a) aluno(a) se ausentou. Em caso de nova ausência, será mantido o conceito "F" para o(a) aluno(a).

#### **AValiação de Recuperação (REC):**

Em cumprimento à Resolução ConsEPE no. 182/2014, todos os alunos que obtiverem conceito final igual a "D" ou "F" terão direito à realização de avaliação de recuperação, que seguirá os seguintes critérios:

- A composição do conceito final após a recuperação será formado segundo a tabela abaixo:

Conceito final antes da REC	REC	Conceito final do quadrimestre
D	A	C
	B	C
	C	D
	D	D
	F	D*
F	A	C
	B	D
	C	D
	D	F
	F	F

\* Para fins de cálculo do conceito final do quadrimestre, garante-se ao aluno o maior conceito entre o obtido antes e após a realização da REC.

#### **PLÁGIOS:**

Por considerar o uso frequente de atividades em computador e internet, a avaliação dos exercícios envolvendo codificação de algoritmos está sujeita a plágios durante o seu desenvolvimento. A fim de preservar o compromisso da universidade com o caráter pedagógico das atividades e o compromisso ético com a propriedade e integridade intelectual, casos suspeitos de plágio serão **severamente** punidos com a **anulação integral de todas as atividades** envolvidas no caso.

**REPROVAÇÃO:**

Em atendimento à Resolução ConsEPE no. 239/2020, só serão lançados na Prograd os conceitos equivalentes à aprovação do(a) aluno(a). Sendo assim, os conceitos iguais a "F" não serão lançados.

**REPROVAÇÃO POR AUSÊNCIAS:**

Atendendo ao disposto na Resolução ConsEPE no. 239/2020, que trata do Programa de Ensino Continuado Emergencial, não haverá reprovação por ausências (item 5 do Anexo I).

**ATIVIDADES DE APOIO (HORÁRIO DE ATENDIMENTO):**

Em cumprimento à Resolução CONSUNI no. 183/2017, esta turma prevê horários de atendimento extraclasse por meio de ferramentas online a serem definidas por cada professor:

- Prof. Marcio K. Oikawa: Atendimento por alguma ferramenta de discussão não-presencial.
- Prof. Fernando Teubl Ferreira

**Referências bibliográficas básicas**

1. Cormen T. H et al., "Algoritmos: Teoria e Prática". Rio de Janeiro: Editora Campus, 2ª edição, 2002
2. Knuth D.E. "The Art of Computer Programming". vols. 1 e 3, Addison-Wesley, 1973;
3. Szwarcfiter, L. Markezon, "Estruturas de Dados e seus Algoritmos". Livros Técnicos e Científicos, 1994;
4. Ziviani N. "Projeto de Algoritmos com implementação em Java e C++". São Paulo: Editora Thomson, 1ª edição, 2007
5. Feofiloff P. "Algoritmos em Linguagem C". Editora Campus/Elsevier, 2009

**Referências bibliográficas complementares**

1. Rodrigues P., Pereira P., Sousa M., "Programação em C++: Algoritmos e Estruturas de Dados", FCA Editora de Informática, 2000;
2. Sedgewick, R. "Algorithms in C++" (Parts 1-4), Addison-Wesley, 3ª edição, 1998;
3. Tenenbaum, A. M., Langsam Y., Augenstein M. J. "Estruturas de Dados Usando C". Editora Pearson Makron Books;
4. Drozdek Adam. "Estrutura de dados e Algoritmos em C++". Thomson Learning, 2002.