

PLANO DE ENSINO **ESTRUTURA DE DADOS**

Implantação 20181

CARGA HORÁRIA: 66h Teórica: 22h Prática: 44h

EMENTA

Explora o conhecimento de soluções clássicas e atuais de problemas por meio de abstração utilizando conjuntos de dados, operações e representações de listas, pilhas, filas e árvores, apoiados em métodos e técnicas, tendo como subsídio uma linguagem de programação.

COMPETÊNCIAS

I. ANALISAR E RESOLVER PROBLEMAS

VI. COMUNICAR-SE ORALMENTE E POR ESCRITO

VIII - PENSAMENTO LÓGICO - Pensar e usar a lógica formal estabelecendo relações, comparações e distinções em diferentes situações.

IX - LÓGICA DE PROGRAMAÇÃO - Dominar comandos lógicos e resultados aplicados à programação.

XVII - DESENVOLVIMENTO EM LINGUAGENS DE PROGRAMAÇÃO - Desenvolver soluções informatizadas por meio de diferentes linguagens de programação conectadas ou não a um sistema de banco de dados. (Sistemas de Informação)

XXII - PROGRAMAÇÃO - Desenvolver algoritmos e modelos para solução de problemas. (Sistemas de Informação)

XVII - PROGRAMAÇÃO - Desenvolver algoritmos e modelos para solução de problemas. (Ciência da Computação)

XVIII - SOLUÇÃO DE PROBLEMAS COMPUTACIONAIS - Solucionar problemas computacionais a partir de preceitos matemáticos e de ferramentas de desenvolvimento. (Ciência da Computação)

OBJETIVOS DE APRENDIZAGEM

Ao final da disciplina o aluno será capaz de em face de um problema,

- Avaliar e propor a melhor estrutura de dados e algoritmo para solucioná-lo;
- Estimar soluções em relação a tempo de desenvolvimento, complexidade computacional e espacial.
- Implementar as estruturas de dados pilha, fila, lista e árvore na resolução de problemas.
- Aplicar os conceitos de alocação estática e dinâmica de memória.
- Identificar a estrutura mais apropriada na resolução de um problema específico.
- Estabelecer as diferenças de aplicação das estruturas pilha, fila, lista e árvores.
- Abstrair estruturas de dados dinâmicas em uma memória computacional.

ATIVIDADE PRÁTICA SUPERVISIONADA			
Objetivos		Atividades a serem desenvolvidas	Avaliação
Explorar o conhecimento de soluções clássicas e atuais de problemas por meio de abstração utilizando conjuntos de dados, operações e representações de listas, pilhas, filas e árvores, apoiados em métodos e técnicas, tendo como subsídio uma linguagem de programação.		As atividades que serão desenvolvidas pelos estudantes encontram-se detalhadas no ambiente virtual de aprendizagem (Blackboard) da disciplina.	Os critérios de avaliação estão explícitos no ambiente virtual de aprendizagem (Blackboard) da disciplina.
CRONOGRAMA DE AULAS			
1 – Conceitos preliminares • Abstração • Linguagens de Programação	Objetivos de Aprendizagem		Competências Relacionadas
	1. Reconhecer as principais características de abstração; 2. Interpretar problemas de abstração do mundo real; 3. Utilizar de abstrações em problemas do mundo real;		I – ANALISAR E RESOLVER PROBLEMAS III – ATINGIR OBJETIVOS VIII - PENSAMENTO LÓGICO;
	Estratégias de Ensino	Avaliação Formativa	Recursos
	Jogo da memória / Pense-pare-compartilhe/ Estudo dirigido Sequência sugerida: 1. Apresentação da disciplina, com os principais desafios no estudo de estruturas de dados. 2. Apresentação dos objetivos de aprendizagem	Ferramentas tecnológicas: utilizar uma ferramenta informatizada, apresentando imagens diversas, e solicitando a abstração das principais características dos alunos;	<i>Artigos sobre abstração em computação:</i> http://www.lbd.dcc.ufmg.br/colecoes/wei/2009/008.pdf https://www.d.umn.edu/~tcolbu/papers/Abstraction.pdf

2 – Tipos Abstratos de Dados	<p>3. Jogo da Memória, com o objetivo de lembrar conceitos de lógica computacional e matemática;</p> <p>4. Atividade pense-pare-compartilhe: os alunos trabalham juntos para responder a um desafio de abstração.</p> <p>5. Propor um estudo dirigido estimulando o aluno a trabalhar com a realização de abstrações, individualmente ou em grupo;</p> <p>6. Utilizar ferramentas tecnológicas para avaliar o entendimento sobre abstração de dados</p>		<p>Quizz: apresentar conceitos de abstração de dados em formato de perguntas e respostas;</p>	
	Objetivos de Aprendizagem	Competências Relacionadas		
<p>2 – Tipos Abstratos de Dados</p> <ul style="list-style-type: none"> • Conceitos • Aplicações 	<p>1. Definir o conceito sobre Tipo Abstrato de Dados.</p> <p>2. Aplicar as características de abstração para o TAD.</p> <p>3. Desenvolver as possíveis operações que cada TAD deve contemplar.</p>		<p>I – ANALISAR E RESOLVER PROBLEMAS</p> <p>III – ATINGIR OBJETIVOS</p> <p>VIII - PENSAMENTO LÓGICO;</p>	
	Estratégias de Ensino	Avaliação Formativa		Recursos
	<p>Aula expositiva dialogada / trabalho</p> <p>Sequência sugerida:</p> <ol style="list-style-type: none"> 1. Apresentação dos objetivos de aprendizagem 2. Levantamento de conhecimentos prévios: diálogo com anotações na lousa. 3. Discussão rápida de levantamento de hipóteses no grupo-sala. 4. Aula expositiva com apoio de ppt. 5. Trabalho em dupla/grupo: propor uma atividade para uma abstração de itens contidos em sala de aula; 6. Apresentação das propostas (chamar apenas alguns grupos de forma aleatória se a sala for muito grande). 7. Feedback organizando e relacionando as abstrações elaboradas pelos estudantes com abstrações válidas no mundo real. 8. Indicação de leitura para a próxima aula. 	<p>Quizz rápido apresentando exemplos de abstrações no ppt.</p>		<p>Capítulo de livro TAD, do professor Waldemar Celes: http://www.ic.unicamp.br/~ra069320/PED/MC102/1s2008/Apostilas/Cap09.pdf</p> <p>Repositório com exemplos de códigos fonte de TAD: https://github.com/malbarbo/na-lp-copl/blob/master/11-tipos-abstratos-de-dados-e-construcoes-encapsuladas.md</p>
	Objetivos de Aprendizagem		Competências Relacionadas	
<p>3 – Tipos Abstratos de Dados</p> <ul style="list-style-type: none"> • Aplicação em Programação 	<p>1. Identificar os principais métodos de programação de TAD</p> <p>2. Aplicar os conceitos de geração e manipulação de TAD em uma linguagem de programação;</p> <p>3. Avaliar as principais estratégias de criação e manipulação de TAD em uma linguagem de programação;</p>		<p>I – ANALISAR E RESOLVER PROBLEMAS</p> <p>III – ATINGIR OBJETIVOS</p> <p>VIII - PENSAMENTO LÓGICO;</p>	
	Estratégias de Ensino	Avaliação Formativa		Recursos
	<p>Aula expositiva dialogada / demonstração / Treino de habilidades</p> <p>Sequência sugerida:</p> <ol style="list-style-type: none"> 1. Apresentação dos objetivos de aprendizagem 2. Levantamento de conhecimentos prévios: diálogo com anotações na lousa. 3. Aula expositiva com apoio de PPT. 4. Aula expositiva dialogada com demonstração: apresentar a estrutura de uma TAD, bem como a implementação de uma operação; (Encorajar os estudantes a revisarem os conteúdos teóricos abordando-os como referências nas atividades práticas e criar um ambiente cooperativo e não competitivo, para que os estudantes se sintam motivados e seguros na demonstração do seu trabalho/conhecimento) 	<p>Feedback formativo: avaliar a implementação de uma TAD pelos alunos, desde a abstração até a implementação de sua estrutura e operações;</p>		<p>Capítulo de livro TAD, do professor Waldemar Celes: http://www.ic.unicamp.br/~ra069320/PED/MC102/1s2008/Apostilas/Cap09.pdf</p> <p>Repositório com exemplos de códigos fonte de TAD: https://github.com/malbarbo/na-lp-copl/blob/master/11-tipos-abstratos-de-dados-e-construcoes-encapsuladas.md</p>
	Objetivos de Aprendizagem		Competências Relacionadas	

<p>4 – Alocação de memória estática</p> <p>Conceitos</p> <p>•</p> <p>Aplicações</p> <p>•</p>	<p>5. Treino de habilidade com um TAD apresentado: com o objetivo de verificar a técnica de programação utilizada pelos alunos e sugerir melhorias;</p> <p>6. Treino de habilidade com um TAD abstraído pelo aluno: com o objetivo de verificar além da técnica de programação utilizada, a abstração feita pelo aluno;</p> <p>7. Indicação de leitura para a próxima aula.</p>		
	Objetivos de Aprendizagem	Competências Relacionadas	
	<p>1. Descrever o procedimento de aplicação de memória estática;</p> <p>2. Analisar diferentes procedimentos de alocação estática de memória em estruturas de dados;</p> <p>3. Projetar estruturas de dados em determinada linguagem de programação com alocação estática de memória;</p>	XVII - PROGRAMAÇÃO	
	Estratégias de Ensino	Avaliação Formativa	Recursos
	<p>Aula expositiva dialogada / demonstração / perguntas e respostas</p> <p>Sequência sugerida:</p> <ol style="list-style-type: none"> 1. Apresentação dos objetivos de aprendizagem 2. Quiz sobre os procedimentos na aula anterior: diálogo com anotações na lousa. 3. Aula expositiva dialogada com apoio de PPT. 4. Aula expositiva com demonstração: exemplos práticos de alocação estática utilizando estruturas lineares; 5. Treino de habilidade com exemplo: apresentar três exemplos, utilizando três estruturas diferentes, demonstrando a alocação estática; 6. Perguntas e respostas utilizando exemplos e contraexemplos de alocação estática de memória; 7. Checklist apresentando exemplos de alocação estática de memória implementadas em uma linguagem de programação, perguntando aos alunos sobre a correte e sobre possíveis melhorias; 8. Indicação de leitura para a próxima aula. 	<p>Checklist: As opções de resposta desta forma de avaliação (<i>checklist</i>) são S (sim) ou N (não); ou opções que indicam se a atividade foi exercida completamente (completa, parcial ou ausente) ou corretamente (total, parcial ou incorreta);</p>	Dependendo da linguagem escolhida para a disciplina, o apoio documental da mesma para alocação estática e dinâmica;
<p>5 – Alocação de memória dinâmica</p> <p>Conceitos</p> <p>•</p> <p>Aplicações</p> <p>•</p>	Objetivos de Aprendizagem	Competências Relacionadas	
	<p>1. Descrever o procedimento de alocação de memória dinâmica;</p> <p>2. Analisar diferentes procedimentos de alocação dinâmica de memória em estruturas de dados;</p> <p>3. Projetar estruturas de dados em determinada linguagem de programação com alocação dinâmica de memória;</p>	XVII - PROGRAMAÇÃO	
	Estratégias de Ensino	Avaliação Formativa	Recursos

<p>6 – Tipo de Dados Lista – Lista linear</p> <ul style="list-style-type: none"> • Conceitos • Aplicação prática 	<p>QUIZ /Aula expositiva dialogada / demonstração / Treino de habilidade/Perguntas e respostas</p> <p>Sequência sugerida:</p> <ol style="list-style-type: none"> 1. Apresentação dos objetivos de aprendizagem 2. Quiz sobre os procedimentos na aula anterior: diálogo com anotações na lousa 3. Aula expositiva com apoio de PPT. 4. Aula expositiva dialogada com demonstração: exemplos práticos de alocação dinâmica utilizando estruturas lineares; 5. Treino de habilidade com exemplo: apresentar três exemplos, utilizando três estruturas diferentes, demonstrando a alocação dinâmica; 6. Perguntas e respostas utilizando exemplos e contraexemplos de alocação dinâmica de memória; 7. Checklist apresentando exemplos de alocação dinâmica de memória implementadas em uma linguagem de programação, perguntando aos alunos sobre a correteude e sobre possíveis melhorias; 8. Indicação de leitura para a próxima aula. 	<p>Checklist: Para avaliação de estruturas de dados implementadas em diversas linguagens de programação, com alocação dinâmica, verificando a correteude;</p>	<p>Dependendo da linguagem escolhida para a disciplina, o apoio documental da mesma para alocação estática e dinâmica;</p>
	Objetivos de Aprendizagem	Competências Relacionadas	
	<ol style="list-style-type: none"> 1. Apresentar o conceito de lista e o encadeamento de elementos; 2. Reconhecer os procedimentos que envolvem listas encadeadas; 3. Construir uma lista encadeada a partir do entendimento do conceito; 4. Analisar operações que compõem uma lista encadeada; 	XVII - PROGRAMAÇÃO	
	<p>Estratégias de Ensino</p> <p>Wiki / Aula expositiva dialogada / demonstração / perguntas e respostas/Treino de habilidades</p> <p>Sequência sugerida:</p> <ol style="list-style-type: none"> 1. Apresentação dos objetivos de aprendizagem 2. Wiki com textos pequenos sobre elementos do mundo real listáveis; 3. Aula expositiva dialogada com apoio de PPT. 4. Demonstração com animações: Utilizar o site https://visualgo.net/pt para demonstrar as operações de uma lista; 5. Treino de habilidade com exemplo de implementação de uma lista encadeada; 6. Perguntas e respostas utilizando exemplos e contraexemplos de listas encadeadas implementadas; 	<p>Avaliação Formativa</p> <p>Checklist: Para avaliação de estruturas de dados do tipo Lista implementadas em diversas linguagens de programação, verificando a correteude;</p> <p>Demonstrações: apresentação de implementação de listas encadeadas em determinada linguagem de programação;</p>	<p>Recursos</p> <p>Capítulo do professor Waldemar Celes sobre listas: http://www.ic.unicamp.br/~ra069320/PED/MC102/1s2008/Apostilas/Cap10.pdf</p> <p>Exemplos e exercícios sobre listas encadeadas: https://www.ime.usp.br/~pf/mac0122-2002/aulas/llists.html</p>

<p>7 – Tipo de Dados Lista - Lista Duplamente Encadeada</p> <ul style="list-style-type: none"> • Conceitos • Aplicação prática 	<p>7. Checklist apresentando exemplos de lista encadeada implementadas em uma linguagem de programação, perguntando aos alunos sobre a corretude e sobre possíveis melhorias;</p> <p>8. Indicação de leitura para a próxima aula.</p>		
	Objetivos de Aprendizagem		Competências Relacionadas
	<p>1. Conceituar lista duplamente encadeada, identificando suas peculiaridades;</p> <p>2. Examinar as principais diferenças para uma lista simplesmente encadeada;</p> <p>3. Construir uma lista duplamente encadeada a partir do entendimento do conceito;</p> <p>4. Analisar operações que compõem uma lista duplamente encadeada;</p>		XVII - PROGRAMAÇÃO
	Estratégias de Ensino	Avaliação Formativa	Recursos
	<p>Brainstorming / Aula expositiva dialogada / treino de habilidade / demonstração / perguntas e respostas/checklist</p> <p>Sequência sugerida:</p> <ol style="list-style-type: none"> 1. Apresentação dos objetivos de aprendizagem 2. Brainstorming sobre as implementações de lista da aula anterior; 3. Aula expositiva dialogada com apoio de PPT. 4. Demonstração com animações: Utilizar o site https://visualgo.net/pt para demonstrar as operações de uma lista duplamente encadeada; 5. Treino de habilidade com exemplo de implementação de uma lista duplamente encadeada; 6. Perguntas e respostas utilizando exemplos e contraexemplos de listas duplamente encadeadas implementadas; 7. Checklist apresentando exemplos de lista duplamente encadeada implementadas em uma linguagem de programação, perguntando aos alunos sobre a corretude e sobre possíveis melhorias; 8. Indicação de leitura para a próxima aula. 	<p>Checklist: Para avaliação de estruturas de dados do tipo Lista implementadas em diversas linguagens de programação, verificando a corretude;</p> <p>Demonstrações: apresentação de implementação de listas encadeadas em determinada linguagem de programação;</p>	<p>Capítulo do professor Waldemar Celes sobre listas: http://www.ic.unicamp.br/~ra069320/PED/MC102/1s2008/Apostilas/Cap10.pdf</p> <p>Exemplos e exercícios sobre listas encadeadas: https://www.ime.usp.br/~pf/mac0122-2002/aulas/llists.html</p>
<p>8 – Tipo de Dados Lista - Lista Circular</p> <ul style="list-style-type: none"> • Conceitos • Aplicação prática 	Objetivos de Aprendizagem		Competências Relacionadas
	<p>1. Reconhecer o conceito de lista circular, identificando suas peculiaridades;</p> <p>2. Examinar as principais diferenças para uma lista simplesmente encadeada;</p> <p>3. Construir uma lista circular a partir do entendimento do conceito;</p> <p>4. Analisar operações que compõem uma lista circular.</p>		XVII - PROGRAMAÇÃO

<p>9 – Tipo de Dados Pilha</p> <ul style="list-style-type: none"> • Conceitos • Operações 			
	Estratégias de Ensino	Avaliação Formativa	Recursos
	Brainstorming / Aula expositiva dialogada / demonstração / perguntas e respostas/treino de habilidade	Checklist: Para avaliação de estruturas de dados do tipo Lista implementadas em diversas linguagens de programação, verificando a corretude;	Capítulo do professor Waldemar Celes sobre listas: http://www.ic.unicamp.br/~ra069320/PED/MC102/1s2008/Apostilas/Cap10.pdf
	Sequência sugerida:	Demonstrações: apresentação de implementação de listas encadeadas em determinada linguagem de programação;	Exemplos e exercícios sobre listas encadeadas: https://www.ime.usp.br/~pf/mac0122-2002/aulas/llists.html
	<ol style="list-style-type: none"> 1. Apresentação dos objetivos de aprendizagem 2. Brainstorming sobre as implementações de lista da aula anterior; 3. Aula expositiva dialogada com apoio de PPT. 4. Demonstração com animações: Utilizar o site https://visualgo.net/pt para demonstrar as operações de uma lista circular; 5. Treino de habilidade com exemplo de implementação de uma lista circular; 6. Perguntas e respostas utilizando exemplos e contraexemplos de listas circulares implementadas; 7. Checklist apresentando exemplos de listas circulares implementadas em uma linguagem de programação, perguntando aos alunos sobre a corretude e sobre possíveis melhorias; 8. Indicação de leitura para a próxima aula. 		
	Objetivos de Aprendizagem	Competências Relacionadas	
	<ol style="list-style-type: none"> 1. Descrever o conceito de pilha, apresentando as regras para manipular a estrutura; 2. Examinar as principais diferenças para uma lista; 3. Construir uma pilha a partir do entendimento do conceito; 4. Analisar operações de inserção e remoção dados; 	XVII - PROGRAMAÇÃO	
	Estratégias de Ensino	Avaliação Formativa	Recursos
	Brainstorming / Aula expositiva dialogada / demonstração / perguntas e respostas / Treino de habilidades	Checklist: Para avaliação de estruturas de dados do tipo Pilha implementadas em diversas linguagens de programação, verificando a corretude;	Material complementar sobre pilhas: https://pt.wikibooks.org/wiki/Algoritmos_e_Estruturas_de_Dados/Pilhas
	Sequência sugerida:	Demonstrações: apresentação de implementação de Pilha em determinada linguagem de programação;	Material complementar sobre pilhas: https://users.ece.cmu.edu/~koopman/stack_computers/index.html
	<ol style="list-style-type: none"> 1. Apresentação dos objetivos de aprendizagem 2. Brainstorming sobre as implementações de lista de aulas anteriores; 3. Aula expositiva dialogada com apoio de PPT. 		

10 – Tipo de Dados Pilha	• Aplicação prática	4. Demonstração com animações: Utilizar o site https://visualgo.net/pt para demonstrar as operações de pilha;		
		5. Treino de habilidade com exemplo de implementação de uma pilha;		
		6. Perguntas e respostas utilizando exemplos e contraexemplos de pilha;		
		7. Checklist apresentando exemplos de pilhas implementadas em uma linguagem de programação, perguntando aos alunos sobre a correção e sobre possíveis melhorias;		
11 – Tipo de Dados Fila	• Conceitos • Operações	8. Indicação de leitura para a próxima aula.		
		Objetivos de Aprendizagem	Competências Relacionadas	
		1. Construir uma pilha a partir do entendimento do conceito; 2. Programar as operações de inserção e remoção de dados; 3. Avaliar implementações de pilha;	XVII - PROGRAMAÇÃO	
		Estratégias de Ensino	Avaliação Formativa	Recursos
10 – Tipo de Dados Pilha	• Aplicação prática	Brainstorming / demonstração / Treino de habilidades / Situação problema	Demonstrações: apresentação de implementação de Pilha em determinada linguagem de programação;	Material complementar sobre pilhas: https://pt.wikibooks.org/wiki/Algoritmos_e_Estruturas_de_Dados/Pilhas
		Sequência sugerida: 1. Apresentação dos objetivos de aprendizagem 2. Brainstorming sobre as implementações de pilha da aula anterior; 3. Demonstração das operações de pilha implementadas em uma linguagem de programação; 4. Treino de habilidade com exemplo de implementação de uma pilha por parte dos alunos, com supervisão e feedback do professor; 5. Situação problema para resolução utilizando pilhas, tais como: Notação Polonesa Inversa, Calculadora para expressões matemáticas; Conversão de número decimal para binário; Retirada de mercadorias de um caminhão de entregas; Mecanismo de fazer/desfazer do Word; Mecanismo de navegação de páginas na Internet (avançar e retornar); 6. Indicação de leitura para a próxima aula.	Situação problema: O objetivo dos problemas propostos na estratégia é suscitar uma discussão produtiva entre os alunos, permitindo que os mesmos produzam soluções baseados nos conceitos de pilha.	Material complementar sobre pilhas: https://users.ece.cmu.edu/~koopman/stack_computers/index.html
		Objetivos de Aprendizagem	Competências Relacionadas	
		1. Descrever o conceito de fila, apresentando as regras para manipular a estrutura; 2. Examinar as principais diferenças para uma lista; 3. Construir uma fila a partir do entendimento do conceito; 4. Analisar operações de inserção e remoção dados;	XVII - PROGRAMAÇÃO	
11 – Tipo de Dados Fila	• Conceitos • Operações	Estratégias de Ensino	Avaliação Formativa	Recursos
		Brainstorming / Aula expositiva dialogada / demonstração / perguntas e respostas / Treino de habilidades	Checklist: Para avaliação de estruturas de dados do tipo Fila implementadas em diversas linguagens de programação, verificando a correção;	Material complementar sobre Filas: https://www.ime.usp.br/~pf/algoritmos/aulas/fila.html
		Sequência sugerida: 1. Apresentação dos objetivos de aprendizagem	Demonstrações: apresentação de implementação de filas em determinada linguagem de programação;	Material complementar sobre aplicações de filas: http://www.ufpa.br/sampaio/curso_de_estdados_1/filas/aplic_pra_ticas.htm

12 – Tipo de Dados Fila	<ul style="list-style-type: none"> • Aplicação prática 	<ol style="list-style-type: none"> Brainstorming sobre as implementações de listas nas aulas anteriores; Aula expositiva dialogada com apoio de PPT. Demonstração com animações: Utilizar o site https://visualgo.net/pt para demonstrar as operações de fila; Treino de habilidade com exemplo de implementação de operações de uma fila; Perguntas e respostas utilizando exemplos e contraexemplos de fila; Checklist apresentando exemplos de filas implementadas em uma linguagem de programação, perguntando aos alunos sobre a correção e sobre possíveis melhorias; Indicação de leitura para a próxima aula. 		
		Objetivos de Aprendizagem	Competências Relacionadas	
		<ol style="list-style-type: none"> Construir uma fila a partir do entendimento do conceito; Programar as operações de inserção e remoção dados; Avaliar implementações de fila; 	XVII - PROGRAMAÇÃO	
		Estratégias de Ensino	Avaliação Formativa	Recursos
13 – Árvores - Árvore Binária	<ul style="list-style-type: none"> • Conceito 	<p>Brainstorming / demonstração / Treino de habilidades / Situação problema</p> <p>Sequência sugerida:</p> <ol style="list-style-type: none"> Apresentação dos objetivos de aprendizagem Brainstorming sobre as implementações de fila da aula anterior; Demonstração das operações de fila implementadas em uma linguagem de programação; Treino de habilidade com exemplo de implementação de uma fila por parte dos alunos, com supervisão e feedback do professor; Situação problema para resolução utilizando fila, tais como: cálculo de distância em grafo utilizando fila; Controle por parte do sistema operacional a recursos compartilhados, tais como impressoras; Buffers de Entrada/Saída; Estrutura de dados auxiliar em alguns algoritmos como a busca em largura; Indicação de leitura para a próxima aula. 	<p>Demonstrações: apresentação de implementação de filas em determinada linguagem de programação;</p> <p>Situação problema: O objetivo dos problemas propostos na estratégia é suscitar uma discussão produtiva entre os alunos, permitindo que os mesmos produzam soluções baseados nos conceitos de pilha.</p>	<p>Material complementar sobre Filas: https://www.ime.usp.br/~pf/algoritmos/aulas/fila.html</p> <p>Material complementar sobre aplicações de filas: http://www.ufpa.br/sampaio/curso_de_estdado_1/filas/aplic_praticas.htm</p>
		Objetivos de Aprendizagem	Competências Relacionadas	
		<ol style="list-style-type: none"> Descrever o conceito de árvore; Descrever o conceito árvore binária; Construir árvore a partir do entendimento do conceito; Analisar operações de caminhamento em árvore; 	XVII - PROGRAMAÇÃO	
		Estratégias de Ensino	Avaliação Formativa	Recursos

14 – Árvores - Árvore Binária de Busca	Conceitos	Aplicações	Brainstorming / Aula expositiva dialogada / demonstração / perguntas e respostas/checklist	<p>Sequência sugerida:</p> <ol style="list-style-type: none"> 1. Apresentação dos objetivos de aprendizagem 2. Brainstorming sobre as implementações de estruturas nas aulas anteriores; 3. Aula expositiva com apoio de PPT. 4. Demonstração com animações: Utilizar o site https://visualgo.net/pt para demonstrar o conceito de árvore binária; 5. Treino de habilidade com exemplo de criação e trajeto de uma árvore binária; 6. Perguntas e respostas utilizando exemplos e contraexemplos de árvores; 7. Checklist apresentando exemplos de árvores binárias implementadas em uma linguagem de programação, perguntando aos alunos sobre a corretude e sobre possíveis melhorias; 8. Indicação de leitura para a próxima aula. 	<p>Checklist: Para avaliação de estruturas de dados de árvores implementadas em diversas linguagens de programação, verificando a corretude;</p> <p>Demonstrações: apresentação de implementação de árvores em determinada linguagem de programação;</p>	<p>Material complementar sobre o TAD árvore: https://www.ime.usp.br/~pf/algoritmos/aulas/bint.html</p> <p>Material complementar sobre o TAD árvore em computação: http://www.noginfo.com.br/arquivos/CC_LP_T13.pdf</p>
			Objetivos de Aprendizagem	Competências Relacionadas		
			<ol style="list-style-type: none"> 1. Descrever o conceito de árvore de busca, apresentando as regras para manipular a estrutura; 2. Conceituar busca a esquerda e busca a direita; 3. Construir uma árvore de busca a partir do entendimento do conceito; 4. Analisar operações de busca em largura e busca em profundidade; 	XVII - PROGRAMAÇÃO		
			Estratégias de Ensino	Avaliação Formativa	Recursos	
			Brainstorming / Aula expositiva dialogada / demonstração / perguntas e respostas/checklist	<p>Sequência sugerida:</p> <ol style="list-style-type: none"> 1. Apresentação dos objetivos de aprendizagem 2. Brainstorming sobre as implementações de estruturas nas aulas anteriores; 3. Aula expositiva com apoio de PPT. 4. Demonstração com animações: Utilizar o site https://visualgo.net/pt para demonstrar o conceito de árvore binária; 5. Treino de habilidade com exemplo de criação e trajeto de uma árvore binária; 6. Perguntas e respostas utilizando exemplos e contraexemplos de árvores; 7. Checklist apresentando exemplos de árvores binárias implementadas em uma linguagem de programação, perguntando aos alunos sobre a corretude e sobre possíveis melhorias; 8. Indicação de leitura para a próxima aula. 	<p>Checklist: Para avaliação de estruturas de dados de árvores implementadas em diversas linguagens de programação, verificando a corretude;</p> <p>Demonstrações: apresentação de implementação de árvores em determinada linguagem de programação;</p>	<p>Material complementar sobre o TAD árvore: https://www.ime.usp.br/~pf/algoritmos/aulas/bint.html</p> <p>Material complementar sobre o TAD árvore em computação: http://www.noginfo.com.br/arquivos/CC_LP_T13.pdf</p>
1	5	1	Objetivos de Aprendizagem	Competências Relacionadas		

16 – Árvores – Árvore AVL	<ol style="list-style-type: none"> 1. Apresentar o conceito e os algoritmos de percurso em árvore; 2. Analisar os algoritmos de percurso em árvore; 3. Compreender o propósito dos algoritmos de percurso em árvore; 		XVII - PROGRAMAÇÃO
	Estratégias de Ensino	Avaliação Formativa	Recursos
	Brainstorming / Aula expositiva dialogada / demonstração / perguntas e respostas/checklist	<p>Infográficos: apresentação de elementos gráfico-visuais, pequenos textos e dados utilizados pelos algoritmos de percurso em árvore;</p> <p>Demonstrações: apresentação de implementação de dos algoritmos de percurso em árvore em determinada linguagem de programação;</p>	<p>Material complementar sobre o percurso em árvores binárias: https://pt.wikibooks.org/wiki/Algoritmos_e_Estruturas_de_Dados/%C3%81rvores_Bin%C3%A1rias</p>
	Sequência sugerida:		
17	Objetivos de Aprendizagem		Competências Relacionadas
	<ol style="list-style-type: none"> 1. Descrever o conceito de árvore AVL; 2. Apresentar o conceito de balanceamento de nós; 3. Construir uma árvore AVL a partir do entendimento do conceito; 4. Analisar operações de balanceamento em árvores AVL; 		XVII - PROGRAMAÇÃO
	Estratégias de Ensino	Avaliação Formativa	Recursos
	Brainstorming / Aula expositiva dialogada / demonstração / perguntas e respostas/checklist	<p>Infográficos: apresentação de elementos gráfico-visuais, pequenos textos e dados utilizados pelos algoritmos de busca, mostrando o seu desempenho;</p> <p>Demonstrações: apresentação de implementação de dos algoritmos de busca percurso em árvore em determinada linguagem de programação;</p>	<p>Material complementar sobre árvores AVL: https://pt.wikibooks.org/wiki/Algoritmos_e_Estruturas_de_Dados/%C3%81rvores_AVL</p>
17	Sequência sugerida:		
	<ol style="list-style-type: none"> 1. Apresentação dos objetivos de aprendizagem 2. Brainstorming sobre as implementações de árvores nas aulas anteriores; 3. Aula expositiva com apoio de PPT. 4. Demonstração com animações: Utilizar o site https://visualgo.net/pt para demonstrar os o conceito de balanceamento; 5. Perguntas e respostas utilizando exemplos e contraexemplos de algoritmos de busca; 6. Infográfico mostrando os métodos de balanceamento; 7. Treino de habilidade com implementações dos métodos de balanceamento disponíveis; 8. Indicação de leitura para a próxima aula. 		

18	Há 4 (quatro) unidades disponíveis para acomodar, não necessariamente nesta ordem: - Aplicação de avaliação da N1 - Revisão ou reforço de conteúdos mediante avaliação da performance da turma - Feriados e eventos fortuitos
19	
20	
21	Prova N2
22	Prova Substitutiva

AVALIAÇÃO

A Nota Final (NF) da disciplina considera os seguintes elementos e valores:

NOTA N1			NOTA N2	
AVALIAÇÃO 1 TEÓRICA	AVALIAÇÃO 2	AVALIAÇÃO 3 PRÁTICA	AVALIAÇÃO 4 APS	AVALIAÇÃO 5 TEÓRICA OU PRÁTICA
Avaliação teórica Individual – avaliação da disciplina com questões objetivas e discursivas – (nota de 0 a 10)	A ser definida pela coordenação do curso: Avaliação Teórica/Prática OU Avaliação Integradora OU Teste de Progresso (nota de 0 a 10)	Avaliação prática Individual ou em grupo com instrumento escolhido de acordo com as competências a serem desenvolvidas – (nota de 0 a 10)	Avaliação da Atividade prática orientada pelo docente, individual ou em grupo. (nota de 0 a 1)	Avaliação individual com instrumento escolhido de acordo com as competências a serem desenvolvidas (nota de 0 a 9)

Onde a Média Final (MF) é calculada através da seguinte média ponderada das duas notas, N1 e N2, com peso, respectivamente de 40% e 60%, resultante da seguinte equação:

$$MF = (N1 * 0,4) + (N2 * 0,6)$$

Para aprovação, a Nota Final deverá ser igual ou superior a 6,0 (seis), além da necessária frequência mínima a 75% das aulas. O estudante que não atingir a média final 6,0 (seis), poderá realizar uma Avaliação Substitutiva, cuja nota substituirá a nota da AVALIAÇÃO 5 obtida, caso seja maior.

BIBLIOGRAFIA BÁSICA

1. ASCENCIO, Ana Fernanda Gomes; ARAÚJO, Graziela Santos. Estrutura de Dados: algoritmos, análise da complexidade e implementações em Java e C/C++ [recurso eletrônico, Biblioteca Virtual Universitária - Pearson]. São Paulo. Pearson. 2010
2. PUGA, Sandra; RISSETTI, Gerson Lógica de Programação e Estruturas de Dados - Com Aplicações em Java - 3ª Ed [recurso eletrônico, Biblioteca Virtual Universitária - Pearson]. Pearson. 2010.
3. TAMASSIA, Roberto; GOODRICH, Michael T., Estruturas de Dados e Algoritmos em Java [recurso eletrônico, Biblioteca Virtual Universitária – Minha Biblioteca]. Porto Alegre: Grupo A, 2011.

BIBLIOGRAFIA COMPLEMENTAR

1. DEITEL, Paul & DEITEL, Harvey. C Como Programar. 6ª Edição [recurso eletrônico, Biblioteca Virtual Universitária - Pearson]. Pearson Prentice Hall, 2011.
2. DEITEL, Paul; DEITEL, Harvey. Java, Como Programar [recurso eletrônico, Biblioteca Virtual Universitária - Pearson]. 8ª ed. São Paulo : Pearson Prentice Hall Brasil, 2010.
3. FORBELLONE, André Luiz Villar. Logica de programação - A construção de algoritmos e estruturas de dados [recurso eletrônico, Biblioteca Virtual Universitária - Pearson]. 3ª Edição. São Paulo – Prentice Hall, 2005.
4. ZIVIANI, Nivio. Projeto de Algoritmos: com implementações em JAVA e C++. Cengage Learning Editores, 03/2012.
5. VELOSO, Paulo Augusto Silva et al. "Estrutura de Dados". 15. ed. Rio de Janeiro: Campus, 1999.

Desenvolvido por	Jonathas Silva dos Santos
Data	12/10/2017
1ª Atualização	
2ª Atualização	

