


**UNIVERSIDADE FEDERAL DE UBERLÂNDIA**
**Faculdade de Computação**

Av. João Naves de Ávila, nº 2121, Bloco 1A - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902

 Telefone: (34) 3239-4144 - <http://www.portal.facom.ufu.br/> [facom@ufu.br](mailto:facom@ufu.br)

**PLANO DE ENSINO**
**1. IDENTIFICAÇÃO**

Componente Curricular:	Algoritmos e Estruturas de Dados 1						
Unidade Ofertante:	Faculdade de Computação (FACOM)						
Código:	GBC024	Período/Série:	2o Período		Turma:		
Carga Horária:				Natureza:			
Teórica:	60h (72 h/a)	Prática:	30h (36 h/a)	Total:	90h (108 h/a)	Obrigatória (X)	Optativa: ( )
Professor(A):	Luiz Gustavo Almeida Martins				Ano/Semestre:	2021-2	
Observações:							

**2. EMENTA**

Alocação dinâmica de memória. Estruturas dinâmicas: pilhas, filas, listas encadeadas. Tipos abstratos de dados.

**3. JUSTIFICATIVA**

O conhecimento dado na disciplina confere ao aluno a capacidade para escolher, especificar e empregar adequadamente as estruturas de dados lineares (listas, filas e pilhas) na implementação de aplicações eficientes, considerando o custo computacional associado e o conceito de tipo abstrato de dados (TAD).

**4. OBJETIVO**
**Objetivo Geral:**

Capacitar os alunos a desenvolverem programas eficientes, em termos de tempo e espaço, utilizando boas práticas de programação, para a solução de problemas baseados em listas, pilhas e filas.

**Objetivos Específicos:**

- Aplicar o conceito de Tipo Abstrato de Dados.
- Discutir o custo computacional de algoritmos.
- Aplicar adequadamente o uso de listas, filas e pilhas.

**5. PROGRAMA**
**1. Revisão da linguagem C:**

- Fundamentos da linguagem
- Tipos de dados (primitivos, indexados e estruturados heterogêneos)

**2. Depuração de programas por meio de um debugger**

### 3. Modularidade em programação:

- Organizar um programa por meio de funções
- Passagem de argumentos para funções: valor e endereço
- Passagem de argumentos para programa: argc e argv

### 4. Ponteiros:

- Operações e aritmética de ponteiros
- Relação entre ponteiros e arrays
- Ponteiros e estruturas

### 5. Alocação Dinâmica de Memória:

- Endereços de memória de um programa: área de código e de dados
- Alocação de endereços pelo compilador
- Alocação de endereços pelo programador: porque é necessário
- Função de alocação dinâmica de memória: retorno da função

### 6. Tipos Abstratos de Dados:

- O que é um Tipo Abstrato de Dados (TADs)
- Porque criar TADs: encapsulamento; reutilização; e manutenção
- Especificação de um TAD
- Como construir TADs: encapsulamento; coesão e acoplamento; organização de funções em vários arquivos-fonte; e geração de código objeto (relocável)

### 7. Listas Lineares:

#### 7.1 - Conceitos básicos:

- O que são listas
- Operações básicas sobre listas
- Especificação do TAD para listas
- Aplicações para uma lista

#### 7.2 - Implementação estática/sequencial:

- Organização contígua: ordenada e não ordenada
- Algoritmos para operações básicas
- Custo dos algoritmos para listas ordenadas e não ordenadas

### 7.3 - Implementação dinâmica/encadeada:

- Organização encadeada: ordenada e não ordenada
- Algoritmos para operações básicas
- Custo dos algoritmos para listas ordenadas e não ordenadas

### 7.4 - Técnicas de encadeamento:

- Uso de nó descritor (nó cabeçalho)
- Encadeamento circular
- Encadeamento duplo

### 7.5 - Listas recursivas

## 8. Pilha

### 8.1 - Conceitos básicos:

- O que é uma pilha
- Política de acesso
- Pilha e recursividade
- Operações básicas sobre pilhas
- Especificação do TAD para uma pilha

### 8.2 - Formas de implementação:

- Pilha estática/sequencial
- Pilha dinâmica/encadeada

### 8.3 - Aplicações com pilha

## 9. Fila

### 9.1 - Conceitos básicos:

- O que é uma fila
- Política de acesso
- Operações básicas sobre filas
- Especificação do TAD para uma fila

## 9.2 - Formas de implementação:

- Fila estática/sequencial
- Fila dinâmica/encadeada

## 9.3 - Aplicações com fila

## 10 - Outras estruturas lineares

- Fila de prioridades
- Deque
- Listas heterogêneas

6. **METODOLOGIA**

O conteúdo da disciplina será ministrado por meio de atividades presenciais e assíncronas. Nas aulas presenciais previstas no calendário acadêmico (94 horas-aula distribuídas em 6 horas-aula semanais), além do conteúdo em si, será trabalhada a aplicação dos conceitos por meio de exemplos e exercícios de fixação. Essas atividades ocorrerão **nas segundas (das 14:50 às 16:30) e quartas (das 14:50 às 16:30 e das 16:50 às 18:30)**. A parte teórica será desenvolvida por meio de aulas expositivas, enquanto a parte prática envolve basicamente roteiros de estudo e exercícios desenvolvidos em laboratório, visando a implementação dos conceitos e estruturas apresentadas. As atividades assíncronas (12 horas-aula distribuídas ao longo do semestre) envolverão o desenvolvimento dos trabalhos da disciplina. A fim de complementar a carga-horária total, ainda está prevista 2 horas-aula presenciais extras na 9ª semana para as apresentações de trabalho, as quais serão previamente agendadas entre o professor e os alunos.

O quadro abaixo apresenta o cronograma das atividades presenciais e assíncronas previstas e sua distribuição em quantidade de horas-aulas durante o semestre letivo:

Semana	Módulos	Atividades Presenciais Previstas	Horas-Aula Presenciais	Data/Hora das Atividades Presenciais	Atividades Assíncronas Previstas	Horas-Aula Assíncronas
02/05/2022	INÍCIO DO SEMESTRE	-	-	-	-	-
1ª	1 e 2	Aula expositiva e exercícios	6	02 e 04/05 - 14:50	-	-
2ª	3	Aula expositiva e exercícios	6	09 e 11/05 - 14:50	-	-

3ª	4 e 5	Aula expositiva e exercícios	6	16 e 18/05 - 14:50	-	-
4ª	6	Aula expositiva e exercícios	6	23 e 25/05 - 14:50	-	-
5ª	7	Aula expositiva e exercícios	6	30/05 e 01/06 - 14:50	-	-
6ª	7	Aula expositiva e exercícios	6	06 e 08/06 - 14:50	Desenvolvimento do 1º trabalho	2
7ª	7	Aula expositiva e exercícios	6	13 e 15/06 - 14:50	Desenvolvimento do 1º trabalho	2
8ª	7	Aula expositiva e exercícios	6	20 e 22/06 - 14:50	Desenvolvimento do 1º trabalho	2
9ª	-	Revisão, prova e apresentação de trabalho	8	27 e 29/06 - 14:50 e 2 h/a a combinar	-	-
10ª	8	Aula expositiva e exercícios	6	04 e 06/07 - 14:50	-	-
11ª	9	Aula expositiva e exercícios	6	11 e 13/07 - 14:50	Desenvolvimento do 2º trabalho	2
12ª	9	Aula expositiva, exercícios e vista de prova	6	18 e 20/07 - 14:50	Desenvolvimento do 2º trabalho	2

13 <sup>a</sup>	10	Aula expositiva e exercícios	6	25 e 27/07 - 14:50	Desenvolvimento do 2º trabalho	2
14 <sup>a</sup>	-	Exercícios, revisão e prova	6	01 e 03/08 - 14:50	-	-
15 <sup>a</sup>	-	Apresentação de trabalho e vista de prova	6	08 e 10/08 - 14:50	-	-
16 <sup>a</sup>	-	Prova de recuperação e Fechamento do semestre	4	17/08 - 14:50	-	-
<b>20/08/2022</b>	<b>TÉRMINO SEMESTRE</b>		<b>Total síncr.:</b> 96 h/a			<b>Total assíncr.:</b> 12 h/a
<b>Carga horária total (síncrona + assíncrona)</b>						108 h/a

### **TÉCNICAS DE ENSINO E FERRAMENTAS TECNOLÓGICAS**

A parte teórica da disciplina será desenvolvida através de aulas expositivas com uso preferencial de recursos audiovisuais (data-show) e exercícios e explicações eventuais com uso do quadro negro e giz. Na parte prática, o conhecimento teórico será complementado através de atividades de implementação em laboratório sob a orientação do professor e monitor, utilizando o ambiente de desenvolvimento (IDE) disponível nos computadores (ex: CodeBlocks) ou disponível on-line (ex: GBD-Online).

Também pretende-se utilizar os ambientes virtuais de aprendizagem MS Teams e Moodle como ferramentas de apoio à disciplina. Uma equipe será criada no MS Teams para facilitar a troca de informação entre o professor e aluno, tanto por meio de postagens quanto via chat, e a entrega das atividades avaliativas (listas e trabalhos). Para ter acesso ao MS Teams, o aluno deve possuir uma conta acadêmica criada a partir de seu e-mail institucional (ufu.br). O link de acesso à equipe da disciplina será enviado por e-mail para os alunos matriculados até a véspera da primeira aula.

O aluno também deve ter uma conta de acesso ao Moodle da UFU, pois todo material de apoio (vídeo-aulas, slides, etc.) será disponibilizado na página da disciplina nesse ambiente virtual, a qual pode ser acessada pelo link:

<https://www.moodle.ufu.br/course/view.php?id=5949>

*OBS: a senha de acesso será disponibilizada em uma postagem na equipe do MS Teams.*

## ATENDIMENTO E COMUNICAÇÃO COM OS DISCENTES

O atendimento aos alunos ocorrerá por meio de um horário fixo extra-classe (**terças das 09:00 às 10:30**) para esclarecimento de dúvidas que possam surgir durante a realização dos trabalhos ou em qualquer outro momento. A troca de mensagens via postagens na equipe do MS Teams também será utilizado, a fim de promover um ambiente colaborativo e coletivo de aprendizagem.

A comunicação com a turma será por meio de postagens na equipe do MS Teams. Excepcionalmente, quando se tratar de assuntos que envolvam apenas um ou um grupo de alunos, o professor pode enviar mensagens via chat ou para o e-mail cadastrado no sistema da UFU.

### 7. AVALIAÇÃO

O método de avaliação pretendido prevê a realização das seguintes atividades avaliativas:

- **Exercícios práticos:** consiste na implementação do conteúdo aprendido na semana. A lista com os exercícios será dada nas aulas práticas. Os exercícios são individuais e devem ser realizados pelo aluno preferencialmente durante a aula, com apoio do professor. Entretanto, será dado o prazo de uma semana para o seu envio, de modo que os alunos ausentes ou que possuem maior dificuldade na implementação tenham tempo de assimilar o conteúdo e completar as atividades. Essa entrega deve ser feita pelo ambiente virtual, conforme especificado na descrição da tarefa. Todos os exercícios serão avaliados em função do correto funcionamento do programa implementado, bem como pelo uso adequado dos conceitos e estruturas de acordo com os requisitos descritos no enunciado. A pontuação máxima que pode ser obtida nessa atividade avaliativa é de **10 pontos**.

- **Trabalhos:** ao longo da disciplina serão realizados dois trabalhos em grupo de 20 pontos (**40 pontos** no total), que envolvem a implementação de operações mais complexas para as estruturas de dados estudadas, bem como o seu uso em aplicações da área de computação. A descrição de cada trabalho será disponibilizada como uma tarefa no ambiente virtual. As respectivas entregas ocorrerão na semana da prova de conteúdo equivalente, de forma que o aluno utilize o desenvolvimento do trabalho como forma de estudo. Essas entregas serão feitas individualmente através do envio de arquivos pelo próprio ambiente de aprendizagem. A correção dos códigos consiste na verificação de seu funcionamento através de casos de testes e da correta implementação e uso das estruturas de dados solicitadas. Além disso, o aluno passará por uma arguição oral individual, em um horário previamente agendado com o professor dentro da semana reservada para esse fim. Essa arguição será realizada presencialmente. A nota da arguição será utilizada como um fator de ponderação (*FP*), entre 0 e 100%, que será aplicado sobre a nota do trabalho (*NT*) para obter a nota do aluno (*NA*). Isto é, a nota do aluno nesta atividade será definida por:  $NA = NT \times FP$ .

- **Provas:** o plano de avaliação prevê a realização de duas provas individuais de 25 pontos (**50 pontos** no total), as quais serão realizadas em sala de aula e consistem em questões acerca do conteúdo apresentado nas aulas e desenvolvidos durante as demais atividades avaliativas. A avaliação será realizada pelo professor a partir da verificação do uso correto e eficiente da estrutura e da forma de implementação solicitada em cada questão.

- **Recuperação:** o plano de avaliação prevê a realização de uma prova individual de 100 pontos como mecanismo de recuperação de aprendizagem aos alunos com frequência mínima de 75% (setenta e cinco por cento) na disciplina e que não atingiram a pontuação necessária para a aprovação, conforme previsto no Art. 141 da Resolução CONGRAD No 46/2022. A prova será realizada em sala de aula, englobará todo o conteúdo da disciplina e sua avaliação será pautada pelo uso correto e eficiente da estrutura de dados e da forma de implementação solicitada em cada questão. A nota final do aluno será a média aritmética entre a nota da prova de recuperação e aquela obtida pelo aluno ao longo do semestre.

A fim de destacar a distribuição dessas atividades ao longo do semestre letivo, é apresentado a seguir o cronograma de avaliação:

<b>Nro</b>	<b>Data</b>	<b>Hora</b>	<b>Descrição</b>	<b>Pontos</b>
1	Entregas semanais entre a 3 <sup>a</sup> e a 14 <sup>a</sup> semanas	Até 14:30	Exercícios	10
2	29/06/2022	14:50 – 16:30	1 <sup>a</sup> prova	25
3	Entrega: 28/06/2022 (envio) Apresentação: previamente agendada para o dia 29/06/2022 das 16:50 às 18:30 ou em algum horário extra-classe (02 horas-aula) na 9a semana	Até 23:59 AGENDADO	1 <sup>o</sup> trabalho	20
4	03/08/2022	14:50 – 16:30	2 <sup>a</sup> prova	25
5	02/08/2022 (envio) Apresentação: previamente agendada para os dias 08/08/2022 das 14:50 às 16:30 e 10/08/2022 das 16:50 às 18:30	Até 23:59 AGENDADO	2 <sup>o</sup> trabalho	20
			TOTAL:	100
6	17/08/2022	14:50 – 16:30	Recuperação	100

A assiduidade nas atividades assíncronas será avaliada mediante a entrega dos respectivos trabalhos.

## 8. BIBLIOGRAFIA

### Básica

- CELES, W., CERQUEIRA, R. & RANGEL, J. L. Introdução a estruturas de dados. Ed. Campus Elsevier, 2004.
- TENENBAUM, A.M., LANGSAM, Y. & AUGENSTEIN, M.J. Estrutura de Dados Usando C. Ed. Makron Books.
- SEDGEWICK, R. & WAYNE, K. Algorithms, 4a ed., Ed. Addison Wesley, 2011.



## **Complementar**

- ZIVIANI, N. Projeto de Algoritmos com implementações em Pascal e C. 3a ed., Ed. Cengage Learning, 2010.
- AHO, A.V., ULLMAN, J.D. & HOPCROFT, J.E. Data Structures and Algorithms. 3a ed., Ed. Addison Wesley.
- CORMEN, T. H., LEISERSON, C. E., RIVEST, R. L. et al. Algoritmos: teoria e prática. 2.ed. Editora Campus, 2002.
- ASCENCIO, A. F. G. & ARAÚJO, G. S. Estrutura de Dados: algoritmos, análise de complexidade e implementações em JAVA e C/C++. Ed. Pearson Prentice Hall, 2010.
- SILVA, O. Q. Estrutura de Dados e Algoritmos Usando C - Fundamentos e Aplicações. Ed. Ciência Moderna, 2007.

## **Material de apoio com acesso remoto:**

- MARTINS, L. G. A. Apostila de Linguagem C: Conceitos Básicos. Material disponibilizado pelo professor no AVA.
- BACKES, A. Linguagem C Descomplicada: Portal de Vídeo-Aulas para Estudo de Programação. <https://programacaodescomplicada.wordpress.com/>, acessado em 22/07/2020.
- Estruturas de Dados Abertas. Disponível gratuitamente em <http://opendatastructures.org/> ou <http://www.facom.ufu.br/~albertini/ods-cpp.pdf>

## 9. DIREITOS AUTORAIS

Todo o material produzido e divulgado pelo docente, como vídeos, textos, arquivos de voz, etc., está protegido pela Lei de Direitos Autorais, a saber, a lei nº 9.610, de 19 de fevereiro de 1998, pela qual fica vetado o uso indevido e a reprodução não autorizada de material autoral por terceiros. Parágrafo Único: responsáveis pela reprodução ou uso indevido do material de autoria dos docentes ficam sujeitos às sanções administrativas e as dispostas na Lei de Direitos Autorais.

## 10. APROVAÇÃO

Aprovado em reunião do Colegiado realizada em: \_\_\_\_/\_\_\_\_/\_\_\_\_

Coordenação do Curso de Graduação: \_\_\_\_\_