



Universidade Federal de Goiás

Instituto de Informática

Ciência da Computação

Matriz Curricular: CICOMP-BI-3 - 2017.1

Plano de Disciplina

Ano Letivo: 2024 - 2º Semestre

#### Dados da Disciplina

Código	Nome	Carga Horária	
		Teórica	Prática
10000063	Introdução à Programação - NBC	48	80

Prof(a): Vinicius da Cunha Martins Borges

Turma: B

#### Ementa

Introdução a algoritmos. Conceitos básicos de programas: constantes; tipos de dados primitivos; variáveis; atribuição; entrada e saída de dados; expressões; estruturas de decisão; estruturas de repetição. Ponteiro. Estruturas de dados homogêneas e heterogêneas: vetores, matrizes, cadeias de caracteres, registros. Subprogramas: funções; passagens de parâmetros por valor e por referência, recursividade. Manipulação de arquivos: abertura, fechamento, leitura e gravação. Tipos de acesso a arquivos: sequencial e indexado. Tipos de arquivos (texto e binário). Transcrição de algoritmos para uma linguagem de programação. Domínio de uma linguagem de programação: sintaxe e semântica; interpretação e compilação de programas; ambiente de desenvolvimento de programas; estilo de codificação; documentação de código; técnicas de depuração e técnicas de profiling; desenvolvimento e uso de bibliotecas.

#### Objetivo Geral

Dar condições para o aluno desenvolver a capacidade de raciocínio lógico e computacional, mapear problemas reais ou abstratos em problemas computacionais e apresentar soluções como algoritmos escritos em uma linguagem de programação.

Habilitar o estudante a definir algoritmos para resolução de problemas básicos e em implementá-los em uma linguagem de programação.

#### Objetivos Específicos

1. Apresentar o estudo da lógica de programação e a construção e implementação de algoritmos coerentes e válidos.
2. Habilitar o estudante ao desenvolvimento de algoritmos para resolver problemas descritos textualmente.
3. Conduzir o estudante a desenvolver algoritmos específicos para a manipulação de problemas numéricos e não-numéricos.
4. Habilitar o estudante a codificar em alguma linguagem de programação os algoritmos desenvolvidos por ele.

#### Relação com Outras Disciplinas

Esta disciplina provê a base para o desenvolvimento de software, a ambientação e a base teórica e prática para o desenvolvimento de programas com uma linguagem de programação. Os conceitos e habilidades de programação desenvolvidos nesta disciplina são básicos para várias outras disciplinas do curso, tais como: Programação Orientada a Objetos, Estruturas de Dados I, Estruturas de Dados II, Compiladores, Sistemas Operacionais 1 e Sistemas Operacionais 2, Computação Gráfica, Redes de Computadores, Sistemas Distribuídos.

#### Programa

1. Conceitos Básicos (Definição de algoritmos e programação de computadores. Estrutura básica de um programa em alguma linguagem de programação. Ambiente de desenvolvimento de programas. Estilos



de codificação. Interpretação/compilação e execução de programas. Tipos primitivos, constantes e variáveis; Expressões aritméticas e lógicas; Comandos de atribuição, de entrada e de saída. 2. Estrutura sequencial e de seleção. 3. Estrutura de Repetição. 4. Estrutura de dados homogêneas – vetores, matrizes e cadeias de caracteres. 5. Estrutura de dados heterogêneas. 6. O tipo ponteiro – definição, aritmética com ponteiros, acesso a elementos de vetores e matrizes através de ponteiros. 7. Funções – declaração, passagem de parâmetros via valor, e por referência, retorno de funções, organização do código com múltiplos arquivos, funções recursivas. 8. Pesquisa sequencial e binária. 9. Manipulação de arquivos: abertura, fechamento, leitura e gravação. Tipos de acesso a arquivos: sequencial e indexado. Tipos de arquivos (texto e binário). 10. Documentação de código; técnicas de depuração e técnicas de profiling; desenvolvimento e uso de bibliotecas. 11. Avaliações de programação em laboratório.

## Procedimentos Didáticos

Legenda	Descrição	Objetivo
AEX	Aula teórica	Transmitir conhecimento utilizando quadro ou slides.
AP	Aula prática	Proporcionar ao aluno a aplicação prática do conteúdo ministrado em aula teórica.
ED	Estudo dirigido	Desenvolver a capacidade analítica, capacidade de síntese, de avaliação crítica e de análise.
OTR	Outros	Transmitir conhecimento utilizando quadro ou slides.
RE	Aula teórica com resolução de exercícios	Desenvolver o raciocínio lógico, criatividade e capacidade de abstração e a capacidade de identificar, analisar e projetar soluções de problemas.
SE	Seminários	Desenvolver o raciocínio lógico, criatividade, capacidade de abstração, capacidade para identificar, analisar, projetar soluções de problemas, a capacidade de comunicação oral e a capacidade de trabalhar em grupo.
TG	Trabalho em grupo	Desenvolver a capacidade de comunicação oral e escrita. Capacidade de trabalhar em grupo.

## Conteúdo Programático / Cronograma

Início	Proc. Didático	Tópico	# Aul.
--------	----------------	--------	--------



Início	Proc. Didático	Tópico	# Aul.
26/08/24	AEX, RE, OTR	<p>PARTE 01</p> <p>ATIVIDADES SUPERVISIONADAS (4,5 + 4,5 horas)</p> <p>Vídeo: Mapa Geral de Ciência da Computação (<a href="https://www.youtube.com/watch?v=SzJ46YA_RaA">https://www.youtube.com/watch?v=SzJ46YA_RaA</a>) Elaboração de texto sobre o vídeo Mapa Geral de Ciência da Computação (Aula 04)</p> <p>Levantamento das contribuições de Charlles Babbage e Ada Lovelace para a Computação Elaboração de texto sobre o assunto</p> <p>AULA 01 Apresentação da disciplina; Plano de ensino; Elementos básicos de um computador; Sistemas e programas usados na disciplina;</p> <p>AULA 02 Conceitos básicos: Definição de Algoritmos e Instruções; Comentários de 1 linha e multilinha; Estrutura básica de um programa em C (regiões de ""Includes"", Programa principal, Declaração de variáveis); Tipos de dados primitivos (Byte e organização da memória; Tipos primitivos; char, short, int, float e double; modificadores de tipos: signed, unsigned e long); Declaração de variáveis de tipos primitivos; Conversão de tipos; Caracteres e Tabela ASCII;</p> <p>AULA 03 Organização inicial de código (Indentação e uso de comentários); Diretivas #define e #include; Compilação e execução via linha de comando (Exemplo de programa em linguagem C); Saída de dados de tipos primitivos ( printf ); Entrada de dados de tipos primitivos ( scanf ); Operador de atribuição; Operadores aritméticos, relacionais, lógicos, unários e binários, operadores de Incremento, precedência de operadores; Estrutura Sequencial;</p> <p>AULA 04 Exercícios (Entrada e saída de dados e conversão de tipos); LISTA 01 - estrutura sequencial e seleção</p> <p>AULA 05 Estrutura de Seleção (if; if/else; if/else aninhados, switch/case);</p> <p>AULA 06 Estrutura de Repetição (while, for e do/while);</p> <p>AULA 07 e 08 Exercícios em sala de aula</p> <p>AULA 09 e 10 Configuração do ambiente para depuração de código Exercícios em sala de aula</p> <p>AULA 11 e 12</p>	40



Início	Proc. Didático	Tópico	# Aul.
26/08/24	AEX, RE, OTR	<p>Exercícios em sala de aula</p> <p>AULA 13 Variáveis locais e globais; Passagem de parâmetros por valor</p> <p>AULA 14 Exemplos com funções;</p> <p>AULA 15 e 16 Exercícios em sala de aula;</p> <p>AULA 17 e 18</p> <p>Exercícios em sala de aula;</p> <p>AULA 19 e 20</p> <p>Prova 01 Funções;</p>	40

Início	Proc. Didático	Tópico	# Aul.
01/10/24	AEX, RE, OTR	<p>PARTE 02</p> <p>ATIVIDADE SUPERVISIONADA (4,5 + 4,5 horas)</p> <p>Pesquisa sobre as contribuições de George Boole e Claude E. Shannon para a Computação</p> <p>Elaboração de texto sobre o assunto</p> <p>John Von Neumann para a Computação</p> <p>Elaboração de texto sobre o assunto</p> <p>AULA 21 e 22</p> <p>Vetores Homogêneos</p> <p>Exemplos de exercícios com vetores;</p> <p>AULA 23 e 24</p> <p>Vetores como parâmetros de funções;</p> <p>Configuração do ambiente para depuração de código</p> <p>Exemplos com vetores e funções;</p> <p>AULA 25 e 26</p> <p>Ponteiros</p> <p>Passagem de parâmetros por referência;</p> <p>AULA 27 e 28</p> <p>Strings</p> <p>Manipulação de strings</p> <p>Biblioteca string.h</p> <p>AULAS 29 e 30</p> <p>Resolução de exercícios em sala de aula</p> <p>AULAS 31 e 32</p> <p>Matrizes;</p> <p>Exemplos de exercícios com Matrizes;</p> <p>AULA 33 e 34</p> <p>Mapeamento de índices;</p> <p>Exercícios com Matrizes e Vetores;</p> <p>AULA 35 e 36</p> <p>Ponteiros;</p> <p>Passagem de parâmetros via referência;</p> <p>Relação de Ponteiros com Vetores e Matrizes;</p> <p>AULA 37 e 38</p> <p>Manipulação de vetores com ponteiros</p> <p>Manipulação de matrizes com ponteiros</p> <p>AULA 39 e 40</p> <p>Alocação dinâmica de memória;</p> <p>Alocação e liberação dinâmica de memória</p> <p>Alocação dinâmica de vetores, matrizes e estruturas (malloc, calloc, free)</p> <p>Exemplos em sala de aula</p> <p>AULAS 41 - 42</p> <p>Exemplos em sala de aula (Vetores, Strings, Matrizes e Alocação dinâmica)</p> <p>AULAS 43 - 44</p> <p>Exemplos em sala de aula (Vetores, Strings, Matrizes e Alocação dinâmica)</p>	52



Início	Proc. Didático	Tópico	# Aul.
01/10/24	AEX, RE, OTR	AULAS 45 - 46 Exercícios  Prova 02: Vetores, Strings, Matrizes e Alocação Dinâmica PARTE 03	52
19/11/24	AEX, RE, OTR	ATIVIDADE SUPERVISIONADA: (4,5 + 4,5 horas)  Pesquisa sobre as contribuições de Alan Turing para a Computação Elaboração de texto sobre o assunto  AULAS 47 - 48 Estruturas Exercícios em sala de aula  AULAS 49 - 50 Ponteiros para Estruturas Alocação Dinâmica de Estruturas Exercícios em sala de aula AULAS 51 e 52 Exercícios em sala de aula  AULAS 53 e 54 Arquivos Tipos de arquivos (Texto e Binário) e acesso (leitura, escrita) fopen, fclose, fread, fwrite, fseek, fprintf, fscanf, fputc, fgetc e feof  AULAS 55 e 56 Exercícios em sala de aula  AULAS 57 e 58 Exercícios em sala de aula  AULAS 59 e 60 Noções de recursividade Exercícios em sala de aula  AULAS 61 e 62 Prova 03  AULAS 63 e 64 Exercícios	36
Total			128

### Critério de Avaliação

O desempenho do aluno na disciplina será quantificado por 3 notas (N1, N2 e N3). Cada avaliação Nx é composta pelas notas de uma Prova Px e listas de exercícios Lx, de forma que

$$N_x = P_x * 0.8 + (L_x) * 0.2$$

A média final (MF) será obtida pela equação:

$$MF = (N1*0.2 + N2*0.45 + N3*0.35)$$

Observações:

1. Para cada prova será considerado todo o conteúdo ministrado até a data de sua aplicação;
2. Será atribuída a nota 0 (zero) a qualquer prova não realizada ou trabalho não entregue na data estipulada;
3. Os alunos que se envolverem em "cola" (colando ou facilitando a cola) ou plágio receberão nota 0 (zero) para a atividade correspondente. O caso poderá ser levado ao conhecimento da Coordenação e do Conselho Diretor do Instituto de Informática para as providências legais;

4. O aluno que não comparecer a pelo menos 75% das aulas estará reprovado por falta;
5. O aluno que não conseguir média final maior ou igual a 6,0 (seis) estará reprovado por média.
6. As provas serão realizadas considerando todo o conteúdo desenvolvido até a data de sua aplicação.
7. O aluno poderá requerer a prova de 2ª chamada, revisão de nota e frequência de acordo com os termos do RGCG vigente.
8. As atividades supervisionadas indicadas no cronograma referem-se às atividades práticas e devem ser desenvolvidas segundo o Art. 16 do RGCG, o qual considera que os cursos presenciais possuem cada hora-aula de 60 (sessenta) minutos, sendo 50 (cinquenta) minutos de aulas teóricas e práticas e 10 minutos de atividades acadêmicas supervisionadas.

#### Data da Realização das Provas

As datas das avaliações constam no cronograma.

Prova 01: 04/10/2024

Prova 02: 22/11/2024

Prova 03: 17/12/2024

#### Local de Divulgação dos Resultados das Avaliações

1. A prévia da nota de cada prova será divulgada assim que o aluno conclui a avaliação utilizando a plataforma Sharif.
2. Ajustes posteriores são possíveis mediante verificação manual dos códigos submetidos.
3. As notas complementares das listas serão divulgadas pela internet e/ou SIGAA.
4. As notas finais serão divulgadas pela internet e/ou SIGAA.
5. O horário de atendimento extraclasse será realizado de forma online ([meet.google.com/usb-cnya-gdt](https://meet.google.com/usb-cnya-gdt)) às quintas-feiras das 15:00 às 16:00.

#### Bibliografia Básica

1. FOBERLONE, A.L.V.; EBERSPACHER, H.F., Lógica de Programação - A construção de algoritmos e estruturas de dados. 3. ed., São Paulo: Prentice Hall, 2005.
2. ASCENCIO, A. F. G.; CAMPOS C, E.A. V. Fundamentos da Programação de Computadores. 3. ed. Editora Pearson, 2010.
3. SCHILDT, H. C Completo e Total. 3ª Ed. São Paulo: Makron Books, 1996.

#### Bibliografia Complementar

1. FEOFILOFF, P. Algoritmos em Linguagem C. Editora Campus/Elsevier, 2009.
2. FARRER, H. et al. Programação Estruturada de Computadores- Algoritmos Estruturados. 3 ed. Rio de Janeiro: LTC, 1989.
3. SEDGEWICK, R. Algorithms in C. 3. ed. Reading, Mass: Addison-Wesley, 1998. ISBN 0201314525.
4. SALVETTI, D.D.; BARBOSA, L.M. Algoritmos, São Paulo: Makron Books, 1998.
5. CORMEN, T. H et al., Algoritmos: Teoria e Prática. 2. ed. Rio de Janeiro: Editora Campus, 2002.

#### Bibliografia Sugerida

- MIZRAHI, V. V. Treinamento em linguagem C++. Módulo I. São Paulo: Makron Books, 2004. - DEITEL, P.; DEITEL, H. C How to Program, 7th ed, 2013.

Termo de Entrega	Termo de Aprovação
Apresentado à Coordenação no dia	Aprovado em Reunião de CD no dia
Prof(a) Vinicius da Cunha Martins Borges Professor	Prof. Dr. Eliomar Araújo de Lima Diretor do Instituto de Informática
Termo de Homologação	
Data de Expedição: Goiânia, ____ de ____ de ____.	

