



**Universidade Federal do Espírito Santo**  
**Centro Tecnológico**  
**Departamento de Informática**

**PLANO DE ENSINO**

<b>Disciplina:</b> Programação II		<b>Código:</b> INF16153	
<b>Carga Horária Semanal:</b> 4h	<b>Teoria:</b> 2h	<b>Exercício:</b> 0h	<b>Laboratório:</b> 2h
<b>Carga Horária Semestral:</b> 60h	<b>Créditos:</b> 3	<b>Período Letivo:</b> 2022/2	

**OBJETIVO:**

Aprender os conceitos básicos/intermediários de programação, usando como ferramenta uma ou mais linguagens de programação pertencentes ao paradigma procedural. Empregar técnicas básicas de compilação em separado e estruturação de programas.

**EMENTA:**

Introdução a tipos abstratos de dados. Modularização de programas: unidades de programação, passagem de parâmetros (valor e referência), separação em módulos, bibliotecas. Manipulação de arquivos: arquivos de texto e binários, acesso sequencial e direto. Alocação dinâmica de memória. Variáveis compostas homogêneas e heterogêneas com alocação dinâmica. Ferramentas de desenvolvimento e técnicas de detecção de erro. Estudo aprofundado de uma ou mais linguagens de programação procedural.

**CONTEÚDO PROGRAMÁTICO:**

- 1 Introdução (4 horas):
  - 1.1 Revisão de conceitos de Programação I
  - 1.2 O que são estruturas de dados (ED)?
  - 1.3 Benefícios e necessidade de ED
  - 1.4 Modelo básico de memória
  - 1.5 Tipos de dados simples e estruturados
- 2 Ferramentas e técnicas de detecção de erros (4 horas)
  - 2.1 Compilação automática com Makefile
  - 2.2 Técnicas de detecção de erro (gdb e depuradores gráficos)
  - 2.3 Testes em tempo de execução (assert)
- 3 Tipos Abstratos de Dados (10 horas):
  - 3.1 Definição de estrutura abstrata
  - 3.2 Pré-condições e pós-condições
  - 3.3 Conceito de software em camadas
- 4 Modularização (6 horas)
  - 4.1 Bibliotecas

- 4.2 Separação em arquivos de cabeçalhos e códigos-fonte
- 4.3 Makefile
- 5 Ponteiros (10 horas)
  - 5.1 Definição e uso
  - 5.2 Passagem de parâmetros
  - 5.3 Funções
  - 5.4 Ponteiro de Funções
- 6 Uso eficiente de memória (16 horas)
  - 6.1 Alocação estática e alocação dinâmica
  - 6.2 Estruturas de dados encadeadas
  - 6.3 Manipulação de ponteiros em C
- 7 Arquivos (4 horas)
  - 7.1 Texto
  - 7.2 Binário
  - 7.3 Acesso Sequencial
  - 7.4 Acesso Direto
- 8 Tópicos avançados (6 horas)
  - 8.1 Tabelas de dispersão em ponteiros de função
  - 8.2 Padrões de projeto em programação estruturada

#### **METODOLOGIA DE ENSINO:**

Para cada conteúdo, serão disponibilizados, via Portal AVA, material contendo apresentações, capítulos de livros disponibilizados on-line e/ou vídeo-aulas selecionadas, além de um conjunto de exercícios. Todas as aulas serão presenciais, dando-se preferência ao uso do laboratório de graduação. Haverá um laboratório para prática de exercícios e um laboratório para resolução de exercícios avaliativos, com periodicidade preferencialmente semanal ou no máximo quinzenal, a depender do conteúdo.

Os exercícios também serão disponibilizados por meio da ferramenta BOCA ou equivalente, que possibilita a auto correção após a submissão de cada exercício. Além disso, haverá resolução de dúvidas por meio de fórum de discussão no AVA ou pela ferramenta Discord. Será disponibilizado um horário de atendimento presencial, havendo disponibilidade de laboratório.

As ferramentas utilizadas nos laboratórios são compiladores C e interfaces de desenvolvimento (IDEs) de uso livre. Para esta disciplina serão utilizados gcc, gdb, valgrind, codeblocks e vscode. Será utilizado o sistema BOCA para resolução de exercícios e o sistema web repl.it.

#### **SISTEMA DE AVALIAÇÃO:**

Lista de Exercícios (L) + dois trabalhos práticos (TP) + uma prova final (PF). Sendo L a soma das notas de cada lista de exercício. Os trabalhos práticos serão avaliados de acordo com o atendimento aos requisitos propostos, clareza de codificação e originalidade.

**Média parcial (MP) calculada por:**  $MP = 0,3 * L + 0,3 * TP1 + 0,4 * TP2$

**A média final (MF) será:**

MF = MP, se  $MP \geq 7,0$

MF =  $(PF + MP)/2$ , se  $MP < 7,0$  (PF: nota da prova final)

MF  $\geq 5,0$  —> aprovado | MF  $< 5,0$  —> reprovado

#### BIBLIOGRAFIA:

<b>BIBLIOGRAFIA BÁSICA</b>	<b>QTD. BIBLIOTECA<sup>1</sup></b>	<b>VERIF.<sup>2</sup></b>
CELES, W; CERQUEIRA, R; RANGEL NETTO, JM. <b>Introdução a estruturas de dados: com técnicas de programação em C.</b> Rio de Janeiro: Campus, 2004., 2004. (Série Editora Campus/SBC). ISBN: 9788535212280	29	2020
KERNIGHAN, Brian W.; RITCHIE, Dennis M. <b>C, a linguagem de programação padrão ANSI.</b> Rio de Janeiro: Campus, 1990	1 [1990] 27 [1989] 2 [1986]	2020
H. Schildt. <b>C Completo e Total</b> , 3ª edição. Pearson / Makron Books, 2008	12 [2008] 2 [2006] 1 [1997]	2020
<b>BIBLIOGRAFIA COMPLEMENTAR</b>	<b>QTD. BIBLIOTECA</b>	<b>VERIF.</b>
VAREJÃO, F.M. <b>Introdução à programação: uma abordagem usando C.</b> Elsevier, 2015.	1	2020
SEDEWICK, R. <b>Algorithms in C.</b> Boston, Mass.: Addison-Wesley, 1998-2002., 1998.	10	2020
ZIVIANI, N. <b>Projeto de algoritmos: com implementações em PASCAL e C.</b> São Paulo, SP: Cengage Learning, 2011., 2011. ISBN: 9788522110506	12	2020
TENENBAUM, AM; LANGSAM, Y; AUGENSTEIN, M. <b>Estruturas de dados usando C.</b> São Paulo, SP: Pearson Makron Books, 2008., 2008. ISBN: 9788534603485.	11	2020
SZWARCFTER, JL; MARKENZON, L. <b>Estruturas de dados e seus algoritmos.</b> Rio de Janeiro: Livros Técnicos e Científicos, c1994., 1994. ISBN: 8521610149	1	2020
CORMEN, TH. <b>Introduction to algorithms.</b> Cambridge, Mass.: New York, N.Y.: The MIT Press; McGraw-Hill, 2009., 2009.	5	2020
KNUTH, DE. <b>The art of computer programming.</b> Upper Saddle River, N.J.: Addison Wesley, 2011., 2011.	1	2020
SEDEWICK, Robert. <b>Algorithms in C.</b> 3rd ed. Boston, Mass.: Addison-Wesley, 1998-2002	10 [1998] 1 [1990]	2015
<b>BIBLIOGRAFIA DISPONÍVEL E ABERTA</b>		
A bibliografia abaixo encontra-se disponível em <a href="https://bibliotecas-digitais.ufes.br/">https://bibliotecas-digitais.ufes.br/</a> Ir em "Plataformas digitais – Minha Biblioteca"		
1. Backes, Andre. <b>Linguagem C - Completa e Descomplicada.</b> 2018. Disponível em: <a href="https://integrada.minhabiblioteca.com.br/#/books/9788595152090/">https://integrada.minhabiblioteca.com.br/#/books/9788595152090/</a>		
2. Celles, Waldemar et al. <b>Introdução a Estruturas de Dados - Com Técnicas de Programação em C.</b> 2016. Disponível em: <a href="https://integrada.minhabiblioteca.com.br/#/books/9788595156654/">https://integrada.minhabiblioteca.com.br/#/books/9788595156654/</a>		

<sup>1</sup> Quantidade reportada pelo site da Biblioteca Central da Ufes (<http://www.bc.ufes.br>), incluindo cópias disponíveis na biblioteca setorial do Centro Tecnológico.

<sup>2</sup> Ano em que a verificação da quantidade em biblioteca foi feita.

3. Cormen, T. **Desmistificando Algoritmos**. 2013. Disponível em: <https://integrada.minhabiblioteca.com.br/#/books/9788595153929/>
4. GUSTEDT, Jens. **Modern C**. Manning Publications, 2019. disp. em [modernc.gforge.inria.fr/](http://modernc.gforge.inria.fr/)
5. GOUGH, Brian J.; STALLMAN, Richard. **An Introduction to GCC**. Network Theory Limited, 2004. disp. em [www.network-theory.co.uk/gcc/intro/](http://www.network-theory.co.uk/gcc/intro/)

**Outras bibliografias online e públicas:**

ASPNES, James. **Notes on Data Structures and Programming Techniques** (CPSC 223, Spring 2021). 2021. disp. em <http://cs.yale.edu/homes/aspnes/classes/223/notes.html>

**Artigos Tópicos avançados em C:**

1. PRESCHERN, Christopher. **Patterns for C Iterator Interfaces**. In: Proceedings of the 22nd European Conference on Pattern Languages of Programs. 2017. p. 1-14.
2. PRESCHERN, Christopher. **C patterns on data lifetime and ownership**. In: Proceedings of the 24th European Conference on Pattern Languages of Programs. 2019. p. 1-13.
3. PRESCHERN, Christopher. **Patterns to escape the# ifdef hell**. In: Proceedings of the 24th European Conference on Pattern Languages of Programs. 2019. p. 1-12.
4. PRESCHERN, Christopher. **Idioms for error handling in C**. In: Proceedings of the 20th European Conference on Pattern Languages of Programs. 2015. p. 1-10.
5. PRESCHERN, Christopher. **Patterns for returning error information in C**. In: Proceedings of the 24th European Conference on Pattern Languages of Programs. 2019. p. 1-14.
6. PRESCHERN, Christopher. **API patterns in C**. In: Proceedings of the 21st European Conference on Pattern Languages of Programs. 2016. p. 1-11.

EM 07/09/2022

**PROF. FILIPE WALL MUTZ**  
E-MAIL: [FILIPE.MUTZ@UFES.BR](mailto:FILIPE.MUTZ@UFES.BR)