

**Caracterização da Disciplina**

Código da disciplina:	<b>MCTA028-15</b>	Nome da disciplina:	<b>Programação Estruturada</b>						
Créditos (T-P-I):	<b>(2-2-4)</b>	Carga horária:	<b>48 horas</b>	Aula prática:	<b>S</b>	Campus:	<b>SA</b>		
Código da turma:	DA1MCTA028-15SA DA2MCTA028-15SA NA1MCTA028-15SA NA2MCTA028-15SA NB1MCTA028-15SA NB2MCTA028-15SA	Turma:	<b>DA1/DA2/NA1/NA2/ NB1 /NB2</b>	Turno:	<b>Diurno/ Noturno</b>	Quadrimestre:	<b>3</b>	Ano:	<b>2022</b>
Docente responsável:		<b>Francisco de Assis Zampirolli/Wagner Tanaka Botelho</b>							

**Alocação da Turma**

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
8:00 - 9:00	<b>DA1/DA2 - TEORIA</b> (Francisco/Francisco)					
9:00 - 10:00	<b>DA1/DA2 - TEORIA</b> (Francisco/Francisco)					
10:00 - 11:00				<b>DA1/DA2 - PRÁTICA</b> (Francisco/Wagner)		
11:00 - 12:00				<b>DA1/DA2 - PRÁTICA</b> (Francisco/Wagner)		
...	...	...	...	...	...	...
19:00 - 20:00	<b>NA1/NA2 - TEORIA</b> (Francisco/Francisco)			<b>NB1/ NB2 - PRÁTICA</b> (Wagner/Francisco)		
20:00 - 21:00	<b>NA1/NA2 - TEORIA</b> (Francisco/Francisco)			<b>NB1/ NB2 - PRÁTICA</b> (Wagner/Francisco)		
21:00 - 22:00	<b>NB1/NB2 - TEORIA</b> (Wagner/Wagner)			<b>NA1/NA2 - PRÁTICA</b> (Francisco/Wagner)		
22:00 - 23:00	<b>NB1/NB2 - TEORIA</b> (Wagner/Wagner)			<b>NA1/NA2 - PRÁTICA</b> (Francisco/Wagner)		

**Ementa**

Apresentar noções básicas e intermediárias sobre algoritmos, programação em linguagens compiladas, compilação, programas em execução (processos), ponteiros, alocação estática e dinâmica de memória, vetores e matrizes, funções e passagem de parâmetros, registros, arquivos e recursividade. Aplicar todos os conceitos apresentados no contexto da resolução de problemas clássicos e novos da computação.

**Estratégias Didáticas a Serem Utilizadas**

- Aulas expositivas presenciais e síncronas dos conteúdos de teoria e prática da disciplina;
- Uso do IDE CodeBlocks na implementação e depuração de códigos na Linguagem C;
- Listas de exercícios com correção automática no Moodle que devem ser resolvidas em C;
- Provas presenciais e síncronas;
- Material complementar, como *notebooks* no Colab, vídeos gravados, entre outros.

**Conteúdo Programático**

Semana	Teoria / Prática	Conteúdo	Estratégias Didáticas	Avaliação
<b>1</b>	<b>Teoria (19/09)</b>	Apresentação da Disciplina / Introdução a Linguagem C (Fundamentos e Organização de Códigos) / Variáveis e Estruturas de Seleção	Aula Expositiva (Slides)	-
	<b>Prática (22/09)</b>	Apresentação da Disciplina / Introdução ao IDE CodeBlocks / Exercícios	Exercícios para Reforçar os Conceitos Estudados na Teoria	Por Meio de Exercícios no Moodle (Correção Automática)
<b>2</b>	<b>Teoria (26/09)</b>	Estruturas de Repetição e Recursão	Aula Expositiva (Slides)	-

	<b>Prática (29/09)</b>	Exercícios	Exercícios para Reforçar os Conceitos Estudados na Teoria	Por Meio de Exercícios no Moodle (Correção Automática)
3	<b>Teoria (03/10)</b>	Vetores e Matrizes	Aula Expositiva (Slides)	-
	<b>Prática (06/10)</b>	Exercícios	Exercícios para Reforçar os Conceitos Estudados na Teoria	Por Meio de Exercícios no Moodle (Correção Automática)
4	<b>Teoria (10/10)</b>	Tipos Definidos Pelo Programador e Arquivos	Aula Expositiva (Slides)	-
	<b>Prática (13/10)</b>	Exercícios	Exercícios para Reforçar os Conceitos Estudados na Teoria	Por Meio de Exercícios no Moodle (Correção Automática)
5	<b>Teoria (17/10)</b>	Ponteiros (Primeira Parte)	Aula Expositiva (Slides)	-
	<b>Prática (20/10)</b>	Primeira Avaliação (P1) – Conteúdo até a Semana 5	-	-
6	<b>Teoria (24/10)</b>	Ponteiros (Segunda Parte)	Aula Expositiva (Slides)	-
	<b>Prática (27/10)</b>	Exercícios Relacionados as Semanas 5 e 6	Exercícios para Reforçar os Conceitos Estudados na Teoria	Por Meio de Exercícios no Moodle (Correção Automática)
7	<b>Teoria (31/10)</b>	Alocação Dinâmica	Aula Expositiva (Slides)	-
	<b>Prática (03/11)</b>	Exercícios	Exercícios para Reforçar os Conceitos Estudados na Teoria	Por Meio de Exercícios no Moodle (Correção Automática)
8	<b>Teoria (07/11)</b>	Ordenação e Listas	Aula Expositiva (Slides)	-
	<b>Prática (10/11)</b>	Exercícios Sobre Ordenação	Exercícios para Reforçar os Conceitos Estudados na Teoria	Por Meio de Exercícios no Moodle (Correção Automática)
9	<b>Teoria (14/11)</b>	Feriado	-	-
	<b>Prática (17/11)</b>	Exercícios sobre Listas	Exercícios para Reforçar os Conceitos Estudados na Teoria	Por Meio de Exercícios no Moodle (Correção Automática)
10	<b>Teoria (21/11)</b>	Filas e Pilhas	Aula Expositiva (Slides)	-
	<b>Prática (24/11)</b>	Exercícios	Exercícios para Reforçar os Conceitos Estudados na Teoria	Por Meio de Exercícios no Moodle (Correção Automática)
11	<b>Teoria (28/11)</b>	Revisão	Aula Expositiva (Slides)	-
	<b>Prática (01/12)</b>	Segunda Avaliação (P2) - Todo Conteúdo	-	-
12	<b>Teoria (05/12)</b>	Prova Substitutiva (SUB) - Será na Sala de Aula e no "Papel"	-	-
	<b>Prática (08/12)</b>	Prova de Recuperação (REC) - Todo o Conteúdo	-	-

**Descrição dos Instrumentos e Critérios de Avaliação Qualitativa**

**Ferramentas:** linguagem de programação C, IDE CodeBlocks e exercícios com correção automática no Moodle;

**Avaliações do Período Letivo Regular:** a disciplina de Programação Estruturada (Teoria e Prática) será composta pelas seguintes provas presenciais: P1 (40%) e P2 (50%). Além disso, Exercícios Práticos (EP) serão disponibilizados no Moodle semanalmente e devem ser resolvidos individualmente, de forma assíncrona e o prazo será de duas semanas. Portanto, a NF será calculada da seguinte forma:

$$NF = 0,4 \times P1 + 0,5 \times P2 + 0,1 \times EP$$

As avaliações serão realizadas nos seguintes dias:

- **P1: 20/10;**
- **P2: 01/12.**

**Conceitos:** com a NF, o conceito da prática será atribuído da seguinte forma:

**9,0 ≤ NP ≤ 10,0 → A**  
**7,5 ≤ NP < 9,0 → B**  
**6,0 ≤ NF < 7,5 → C**  
**5,0 ≤ NF < 6,0 → D**  
**NF < 5,0 → F**

**Avaliação Substitutiva (SUB):** estarão habilitados os discentes que se ausentarem a uma das avaliações do período regular e contemplados pelo benefício de acordo com a Resolução CONSEPE no. 227, de 23 de abril de 2018. O discente deve solicitar a SUB com pelo menos 24h antes da avaliação.

- **Data da SUB: 05/12 (Importante ressaltar que a SUB será na sala de aula e no “papel”).**

**Avaliação de Recuperação (REC):** estarão habilitados para a avaliação de recuperação os alunos que obtiverem conceito final D ou F na conclusão de todas as atividades e avaliações aplicadas no período letivo regular, obedecendo as regras indicadas na Resolução CONSEPE no. 182, de 23 de outubro de 2014. Portanto, o Conceito Final na disciplina após a REC será definido da seguinte forma:

Conceito (Pré-REC)	Conceito REC	Conceito Final
<b>D</b>	<b>A</b>	<b>C</b>
	<b>B</b>	<b>C</b>
	<b>C</b>	<b>C</b>
	<b>D</b>	<b>D</b>
	<b>F</b>	<b>D</b>

Conceito (Pré-REC)	Conceito REC	Conceito Final
<b>F</b>	<b>A</b>	<b>C</b>
	<b>B</b>	<b>C</b>
	<b>C</b>	<b>D</b>
	<b>D</b>	<b>F</b>
	<b>F</b>	<b>F</b>

- **Data da REC: 08/12.**

**IMPORTANTE!!**

**Presença:** as aulas terão listas de presenças e dúvidas não serão respondidas por email para os alunos que não estiverem presentes nas aulas sem justificativa legal.

**Provas:** as provas serão individuais e sem consulta. O aluno que quiser poderá levar uma folha A4 manuscrita (somente um lado) com qualquer informação da sintaxe da linguagem C. Contudo, códigos de resolução de exercícios não serão permitidos. Além disso, a folha deve ser assinada e entregue junto com a prova.

**Plágio:** copiar o código de colegas e fazer pequenas alterações (em comentários, em espaçamentos, etc) **É PLÁGIO**. Porém, tirar dúvidas com colegas **NÃO É PLÁGIO**.

De acordo com o Regimento Geral da UFABC, artigo 77, o dever dos membros de corpo discente é “*comportar-se de acordo com os princípios éticos*”. Ou seja, cada aluno deve enviar para avaliação apenas o resultado de seu **PRÓPRIO TRABALHO**.

Em particular, enviar para avaliação algo que não foi feito por si mesmo (e assim assumir a sua autoria) é considerado plágio e, portanto, eticamente inaceitável.

De acordo com o Código de Ética da UFABC, artigo 25, disponível no seguinte *link*: <https://abre.ai/e2t8>

"é eticamente inaceitável que os discentes:

- I. fraudem avaliações;
- II. fabriquem ou falsifiquem dados;
- III. plagiem ou não creditem devidamente autoria;
- IV. aceitem autoria de material sem participação na produção;
- V. vendam ou cedam autoria de material acadêmico próprio a pessoas que não participaram da produção."

Portanto, qualquer violação às regras implicará:

- Descarte dos conceitos atribuídos a **TODAS** as tarefas avaliativas regulares de **TODOS** os envolvidos, causando, assim, suas **REPROVAÇÕES AUTOMÁTICAS** com conceito **F** na **DISCIPLINA** e sem a possibilidade de realizar a **REC**;
- Possível denúncia à Comissão de Transgressões Disciplinares Discentes da Graduação, a qual decidirá sobre a punição adequada à violação que pode resultar em advertência, suspensão ou desligamento, de acordo com os artigos 78-82 do Regimento Geral da UFABC;
- Possível denúncia apresentada à Comissão de Ética da UFABC, de acordo com o artigo 25 do Código de Ética da UFABC.

Sendo assim, o discente está automaticamente **REPROVADO** na disciplina, sem a possibilidade de realizar a **REC**, caso seja identificado **PLÁGIO** nas **Listas de Exercícios do Moodle** ou em uma das **Provas** (P1 ou P2).

#### Horário de Atendimento aos Alunos (Síncrono)

Em cumprimento à Resolução CONSUNI no. 183/2017, o seguinte dia/horário será utilizado para o atendimento aos discentes:

- **Prof. Francisco de Assis Zampiroli**: quinta-feira, das 14:00 às 16:00, na sala 819/Bloco B em Santo André;
- **Prof. Wagner Tanaka Botelho**: quinta-feira, das 14:00 às 16:00, na sala 804/Bloco B em Santo André.

#### Referências Bibliográficas Básicas

1. PINHEIRO, F. A. C. Elementos de programação em C. Porto Alegre, RS: Bookman, 2012.
2. FORBELLONE, A. L. V.; EBERSPACHER, H. F. Lógica de programação: a construção de algoritmos e estruturas de dados. 3ª edição. São Paulo, SP: Prentice Hall, 2005.
3. CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Algoritmos: teoria e prática. 2ª edição. Rio de Janeiro, RJ: Campus, 2002.

#### Referências Bibliográficas Complementares

1. AGUILAR, L. J. Programação em C++: algoritmos, estruturas de dados e objetos. São Paulo, SP: McGraw-Hill, 2008.
2. DROZDEK, A. Estrutura de dados e algoritmos em C++. São Paulo, SP: Cengage Learning, 2009.
3. KNUTH D. E. The art of computer programming. Upper Saddle River, USA: Addison-Wesley, 2005.
4. SEDGEWICK, R. Algorithms in C++: parts 1-4: fundamentals, data structures, sorting, searching. Reading, USA: Addison-Wesley, 1998