



UNIVERSIDADE FEDERAL DO ABC – UFABC
CENTRO DE MATEMÁTICA, COMPUTAÇÃO E COGNIÇÃO
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

PLANO DE ENSINO

para o Quadrimestre Suplementar (QS) – Resolução CONSEPE 240/2020

ANO LETIVO	QUADRIMESTRE	TURNO	CAMPUS
2020	QS	Diurno	Santo André

CÓDIGO	NOME
MCTA028-15SA	Programação Estruturada
Turma	RECOMENDAÇÃO
DAMCTA028-15SA	Processamento da Informação

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.

SOBRE DA DISCIPLINA

A disciplina será conduzida no ambiente virtual de aprendizagem (AVA) Moodle UFABC:

<https://moodle.ufabc.edu.br/>

Curso: **DAMCTA028-15SA - Programação Estruturada - Paulo Henrique Pisani - 2020.QS**

Fique atento aos comunicados realizados no Moodle UFABC, que será utilizado para a condução da disciplina.

Mapa de Atividades

Disciplina: Programação Estruturada – T-P-I 2-2-4

Docente: Paulo Henrique Pisani

Quadrimestre: 2020.QS

Descrição das atividades no mapa de atividades:

- **Horas:** a coluna horas na tabela é soma de T+P. Contudo, pode ser necessário tempo adicional para estudo, conforme indicado no T-P-I da disciplina.
- **Assistir videoaulas (gravadas):** serão disponibilizados links para videoaulas sobre o conteúdo da disciplina. É necessário um navegador web com capacidade de executar vídeos armazenados no Google drive. Durante as videoaulas, além de apresentar o conteúdo, podem ser realizados exemplos práticos de implementação dos conceitos. Ao longo dos vídeos, poderão ser propostos exercícios (não avaliativos).
- **Ler material:** será disponibilizado material para leitura (por exemplo, slides da aula em PDF ou outro material sobre o conteúdo da disciplina). É recomendável ler o material, além de assistir às videoaulas.
- **Exercícios para submissão pelo Moodle:** serão disponibilizados exercícios ao longo do quadrimestre. Todos os exercícios são individuais (devem ser resolvidos pelo próprio aluno). Os exercícios deverão ser submetidos pelo Moodle. Cada exercício pode ter pontuação máxima e prazo de entrega diferentes (essas informações estarão em cada exercício no Moodle). Os exercícios de programação para nota possuem a indicação [EP] no Moodle.

Avaliação dos exercícios de programação:

Poderá ser utilizado um sistema de correção automática. A princípio, a correção automática (caso esteja disponível) aplicará o seguinte critério:

- *Exercícios com pontuação máxima até 1 ponto:* é necessário passar em todos os casos de teste para obter 1 ponto. Caso contrário, ficará com zero.
- *Exercícios com pontuação máxima superior a 1 ponto:* se não passar nos casos de teste abertos (que tem entrada e saída disponíveis/abertas), ficará com zero. Se passar nos casos de teste abertos, a nota poderá ser maior que zero. Dessa forma, estes exercícios podem receber pontuação parcial.
- Em todos os casos, poderá ser aplicado um desconto de 20% da pontuação máxima caso toda a memória alocada não seja liberada pelo programa (vazamento de memória - *memory leak*).
- **Importante:** o professor poderá atribuir nota manualmente nos exercícios e alterar a nota atribuída pelo sistema de correção automática também (por exemplo, devido a pontos que o corretor automático não pode identificar ou problemas/erros na avaliação).

Semana	Horas (T+P)	Tema / Subtema	Objetivos específicos	Atividades teóricas	Atividades práticas
01 21/09/2020 a 27/09/2020	4h	Introdução à linguagem C <ul style="list-style-type: none"> • Compilação • Tipos de dados • Entrada/Saída • Operadores aritméticos, relacionais, lógicos • Estrutura condicional • Estruturas de repetição • Estilo de codificação 	Aplicar a linguagem C para implementação de programas, incluindo os programas propostos na disciplina.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					
02 28/09/2020 a 04/10/2020	4h	Vetores, Funções e passagem de parâmetros <ul style="list-style-type: none"> • Vetores • Funções • Passagem de parâmetros 	Entender os conceitos de programação estruturada até esta semana. Aplicar a linguagem C nos problemas propostos na disciplina, implementando com uso dos conceitos de programação estruturada apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					
03 05/10/2020 a 11/10/2020	4h	Strings, Matrizes <ul style="list-style-type: none"> • Strings • Matrizes 	Entender os conceitos de programação estruturada até esta semana. Aplicar a linguagem C nos problemas propostos na disciplina, implementando com uso dos conceitos de programação estruturada apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					

04 12/10/2020 a 18/10/2020	4h	Recursão	Entender os conceitos de programação estruturada até esta semana. Aplicar a linguagem C nos problemas propostos na disciplina, implementando com uso dos conceitos de programação estruturada apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					
05 19/10/2020 a 25/10/2020	4h	Ponteiros - parte 1 <ul style="list-style-type: none"> • Endereçamento de memória • Ponteiros • Alocação estática e alocação dinâmica 	Entender os conceitos de programação estruturada até esta semana. Aplicar a linguagem C nos problemas propostos na disciplina, implementando com uso dos conceitos de programação estruturada apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					
06 26/10/2020 a 01/11/2020	4h	Ponteiros - parte 2 <ul style="list-style-type: none"> • Alocação dinâmica (vetores) • Aritmética de ponteiros 	Entender os conceitos de programação estruturada até esta semana. Aplicar a linguagem C nos problemas propostos na disciplina, implementando com uso dos conceitos de programação estruturada apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					

07 02/11/2020 a 08/11/2020	4h	Ponteiros - parte 3 <ul style="list-style-type: none"> • Alocação dinâmica (matrizes) • Ponteiro para função 	Entender os conceitos de programação estruturada até esta semana. Aplicar a linguagem C nos problemas propostos na disciplina, implementando com uso dos conceitos de programação estruturada apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					
08 09/11/2020 a 15/11/2020	4h	Estruturas e arquivos	Entender os conceitos de programação estruturada até esta semana. Aplicar a linguagem C nos problemas propostos na disciplina, implementando com uso dos conceitos de programação estruturada apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					
09 16/11/2020 a 22/11/2020	4h	Listas ligadas - parte 1	Entender os conceitos de programação estruturada até esta semana. Aplicar a linguagem C nos problemas propostos na disciplina, implementando com uso dos conceitos de programação estruturada apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					

10 23/11/2020 a 29/11/2020	4h	Listas ligadas - parte 2	Entender os conceitos de programação estruturada até esta semana. Aplicar a linguagem C nos problemas propostos na disciplina, implementando com uso dos conceitos de programação estruturada apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					
11 30/11/2020 a 06/12/2020	4h	Ordenação Simples, Busca: <ul style="list-style-type: none"> Algoritmos de ordenação simples (<i>selection sort</i>, <i>bubble sort</i>, <i>insertion sort</i>) Algoritmos de busca (sequencial e binária) 	Entender os conceitos de programação estruturada até esta semana. Aplicar a linguagem C nos problemas propostos na disciplina, implementando com uso dos conceitos de programação estruturada apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	Exercícios para submissão pelo Moodle.
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					
12 07/12/2020 a 12/12/2020	4h	Revisão	Entender os conceitos de programação estruturada até esta semana. Aplicar a linguagem C nos problemas propostos na disciplina, implementando com uso dos conceitos de programação estruturada apresentados até esta semana.	- Assistir videoaulas (gravadas) - Ler material	
Feedback, comunicação e avaliação (exercícios de programação): avaliação dos exercícios.					

AVALIAÇÃO

Avaliação do Período Letivo Regular: a média final será composta por exercícios de programação passados ao longo do quadrimestre. Cada exercício pode ter pontuação máxima diferente assim como prazos de entrega diferentes. Os exercícios de programação para nota possuem a indicação [EP] no Moodle. Exercícios não entregues no prazo receberão nota zero. Todos os exercícios são individuais (devem ser resolvidos pelo próprio aluno). Será utilizada ferramenta para identificação de submissões com alta similaridade no Moodle.

A média final (MF) será a média ponderada da pontuação obtida em todos os exercícios:

$$MF = 10 \times \frac{\text{soma da pontuação obtida nos exercícios (EPs)}}{\text{soma da pontuação máxima dos exercícios (EPs)}}$$

A partir da média final (MF), o conceito final (CF) será obtido por meio da seguinte tabela:

Conceito Final (CF)	Média Final (MF)
A	$MF \geq 9$
B	$9 > MF \geq 7,5$
C	$7,5 > MF \geq 6$
D	$6 > MF \geq 5$
F	$MF < 5$
O	(reprovação por ausência de avaliação)

Mecanismo de Recuperação (REC):

Estarão habilitados para o mecanismo de recuperação os alunos que obtiverem conceito final **D** ou **F**, obedecendo as regras indicadas na Resolução CONSEPE no. 182 de 23 de outubro de 2014.

Como mecanismo de recuperação, o aluno terá a possibilidade de submeter exercícios passados ao longo do quadrimestre. A nota obtida em cada exercício entregue substituirá a nota previamente obtida naquele exercício. A submissão será por meio de atividade para recuperação no Moodle.

Prazo para entrega dos exercícios (recuperação): **18/12/2020**.

Após a substituição das notas dos exercícios entregues, a média final com recuperação (MR) será obtida com a mesma fórmula usada para a média final (considerando todos os exercícios passados do quadrimestre). Será aplicada a tabela de conversão de conceito a seguir (o conceito final com recuperação será no máximo C):

Conceito Final com Recuperação (CFR)	Média Final com Recuperação (MR)
C	$MR \geq 6$
D	$6 > MR \geq 5$
F	$MR < 5$

Caso seja identificado plágio em qualquer exercício entregue, será atribuído conceito final F.

FERRAMENTAS

Para acompanhar a disciplina é necessário computador com acesso à internet:

- Recomenda-se o uso do sistema operacional Ubuntu Linux 18.04 64-bit;
- Caixa de som ou fone de ouvido;
- Navegador web compatível com o Moodle UFABC e Google Meet; O navegador web deve permitir a execução de vídeos a partir do Google drive.
- Editor de texto (por exemplo, gedit, Notepad++, etc).
- Compilador C: gcc - recomenda-se a versão disponível no Ubuntu Linux 18.04 64-bit
- Opcional: valgrind, GDB (GNU Debugger), e Code::Blocks 20.03

ATIVIDADES DE APOIO

O professor estará disponível nos seguintes dias/horários para atendimento aos alunos (dúvidas sobre o conteúdo da disciplina):

- **terça-feira: das 15h às 17h** – atendimento por chat ou por e-mail;
- **quinta-feira: das 10h às 12h** – atendimento por videoconferência.

Para o atendimento por videoconferência, será utilizado o Google Meet (a ferramenta poderá ser outra a critério do professor). Haverá link para acesso disponível na página da disciplina no Moodle. Para o atendimento por chat, será usado o chat do Moodle.

Em caso de feriados, o atendimento não será síncrono. Contudo, será possível enviar dúvidas para resposta em um momento posterior (de forma assíncrona). O chat do Moodle, por exemplo, permite o envio de mensagens de forma assíncrona também.

O atendimento síncrono depende de disponibilidade de internet. Em caso de indisponibilidade de internet no momento do atendimento, a resposta para as dúvidas ocorrerá de forma assíncrona. Os alunos podem enviar suas perguntas por e-mail ou pelo chat do Moodle e o professor poderá responder assim que possível.

BIBLIOGRAFIA

Bibliografia básica:

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

Bibliografia complementar:

- AGUILAR, L. J. Programação em C++: algoritmos, estruturas de dados e objetos. São Paulo, SP: McGraw-Hill, 2008.
- DROZDEK, A. Estrutura de dados e algoritmos em C++. São Paulo, SP: Cengage Learning, 2009.
- KNUTH D. E. The art of computer programming. Upper Saddle River, USA: Addison-Wesley, 2005.

- SEDGEWICK, R. Algorithms in C++: parts 1-4: fundamentals, data structures, sorting, searching. Reading, USA: Addison-Wesley, 1998.
- SZWARCFITER, J. L.; MARKENZON, L. Estruturas de dados e seus algoritmos. 3. ed. Rio de Janeiro, RJ: LTC, 1994.
- TENENBAUM, A. M.; LANGSAM, Y.; AUGENSTEIN, M. J. Estruturas de dados usando C. São Paulo, SP: Pearson Makron Books, 1995.

PROFESSOR(ES) RESPONSÁVEL(IS)

Prof. Dr. Paulo Henrique Pisani