



Programação Estruturada

Terceiro quadrimestre de 2021

Prof. Daniel Morgato Martin

Prof. Jesús Pascual Mena Chalco

Centro de Matemática, Computação e Cognição

Universidade Federal do ABC

Av. dos Estados, 5001

Santo André - SP

CEP 09210-580

Email: daniel.martin@ufabc.edu.br, favor usar o prefixo [2021 . PE] no assunto do email.

O prof. Jesús irá conduzir o curso, mais precisamente os plantões de dúvidas, possivelmente a partir do final do mês de Setembro, quando o prof. Daniel sairá de licença. Este plano de ensino foi elaborado de comum acordo entre os dois professores. As informações para contato serão atualizadas quando isso de fato ocorrer.

Objetivos

Objetivos gerais: que o aluno adquira familiaridade com uma linguagem de programação estruturada de nível intermediário com gerenciamento explícito de memória. Que o aluno aprofunde suas habilidades de modelagem e implementação de algoritmos para solucionar problemas computacionais.

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.

* Retirada da página 1030 do [catálogo de disciplinas 2019-2020](#) da UFABC.

Bibliografia

1. KERNIGHAN, Brian W.; RITCHIE, Dennis M.. The C programming language. 2. Ed. Englewood Cliffs, USA: Prentice Hall, c1988. xii, 272. (Prentice-Hall software series). ISBN 9780131103627.
2. SEDGEWICK, Robert. Algorithms in C, vol. 1-4: part 1-4: fundamentals, data structures, sorting, searching. 3. ed. Reading, USA: Addison-Wesley, c1998. Xvii, 702. ISBN 9780201756081.
3. PINHEIRO, Francisco A. C. Elementos de programação em C. Porto Alegre, RS: Bookman, 2012. xx, 528 p., il. ISBN 9788540702028.
4. FEOFILOFF, Paulo. Algoritmos em linguagem C. Rio de Janeiro, RJ: Elsevier, 2009. 208 p. ISBN 9788535232493.

5. TANENBAUM, A. M.; LANGSAM, Y.; AUGENSTEIN, M. J. Estruturas de dados usando C. São Paulo, SP: Pearson Makron Books, 1995.

Carga de trabalho e recomendações

A carga de trabalho semanal estimada para a oferta tradicional desta disciplina é de 2 horas com aulas teóricas, 2 horas com aulas práticas e 4 horas de estudos individuais. Esta disciplina possui uma disciplina recomendada: Processamento da Informação. Eventuais deficiências nessa disciplina irão impactar diretamente no tempo de estudo necessário para você passar no curso.

Regras de oferecimento

O oferecimento desta disciplina no quadrimestre suplementar seguirá as regras definidas no Anexo I da Resolução nº 240/2020 do CONSEPE (veja a página 11 do [Boletim de Serviço Nº 963 de 15 de julho de 2020](#)).

Calendário*

Período regular de aulas	12 semanas	13 de setembro a 03 de dezembro
Período de recuperação	1 semana	Entregar REC no 15 de dezembro

* Veja o [Calendário Acadêmico da Graduação 2021](#) da UFABC para mais detalhes.

Ambiente virtual de aprendizagem (AVA)

Usaremos o site [2021.PE.Q3 no Moodle do CMCC](#) para divulgar o material com o conteúdo do curso, links para as videoaulas, links para a entrega das listas de exercício semanais (com correção automática) e para entrega de outras atividades avaliativas que eventualmente venham a ser realizadas.

Mapa de atividades e abordagem do curso

Todas as semanas o professor irá disponibilizar no AVA os seguintes materiais e atividades:

- videoaula com explicações da teoria do tópico previsto para aquela semana;
- videoaula com atividades guiadas pelo professor, com enunciados, dicas e resoluções de exercícios;
- tarefa de leitura para que o aluno complemente e afixe na memória o que foi visto na aula teórica;
- **(para nota)** exercício-programa (EP) para o aluno implementar em casa;
- ocasionalmente, algum material extra.

O aluno deverá entregar os EPs que foram disponibilizados, obedecendo aos prazos determinados, e que serão corrigidos automaticamente pelo módulo VPL do Moodle.

Tirando suas dúvidas

O professor fará plantões via Google Meet, toda semana, nos seguintes horários:

Dia / Período	Matutino	Noturno
Terças-feiras	8:00 -- 9:30	
Sexta-feiras	8:00 -- 9:30	

O link para o plantão ficará disponível na página inicial do nosso site no Moodle. Você também pode enviar suas dúvidas pelo Telegram, para **@danielmmartin**, ou para o email daniel.martin@ufabc.edu.br, colocando o prefixo [2021.PE] no assunto da mensagem.

Código de honra

Será adotado o código de honra descrito na [página mantida pelo prof. Emilio Franceschini](#). Uma versão em PDF pode ser encontrada na primeira seção do nosso site do Moodle. **Resumindo: se você colar em alguma atividade avaliativa, você receberá 0 em todas as atividades avaliativas regulares do curso, sejam elas a presente, as passadas ou as futuras e estará sumariamente reprovado com média final 0. No entanto, poderá fazer a REC se assim desejar.**

Avaliações

Exercícios-programas

O professor irá disponibilizar o material de estudo semanalmente: as videoaulas (teórica e prática), a tarefa de leitura e o exercício-programa (EP) -- às vezes mais de um -- que você deverá entregar resolvido (via Moodle) até o prazo indicado. Você conseguirá entregar o exercício com até 48h de atraso. **Não serão contados para nota os programas entregues depois deste prazo adicional!** Se o Moodle ou o módulo VPL (que faz a correção automática dos programas) estiver indisponível durante o prazo regular, você deve reportar esse fato para o professor, imediatamente, para que o problema seja resolvido. Se o site vier a falhar durante as 48h seguintes ao prazo regular, nenhum prazo será estendido e você não deve esperar que o professor resolva o problema a tempo de você entregar o programa atrasado.

O módulo VPL

As atividades de implementação que chamamos de exercícios-programas (EPs) serão corrigidas automaticamente pelo módulo VPL do Moodle. Você submete seu código fonte na linguagem C, que será a linguagem usada na disciplina, e o módulo VPL compila seu programa e (se ele compilar corretamente) executa uma bateria de 100 testes e verifica, para cada uma das 100 entradas de teste, se a saída do seu programa é igual à saída esperada. A sua nota será proporcional ao número de testes em que seu programa passar. Você pode submeter quantas vezes quiser, até obter nota máxima. Porém só será contada na média final a última nota obtida dentro do prazo regular ou nas 48h seguintes.

O módulo VPL conta com um sistema de detecção automática de plágio que compara seu programa com os dos colegas. Ele pode (e será) alimentado com programas entregues por alunos em ofertas anteriores desta disciplina. **Resumindo: não tente colar, pois você será pego com alta probabilidade!**

Compto da média final (MF)

A média final será a média aritmética das notas obtidas nas atividades (EPs) atribuídas ao longo do quadrimestre, **desprezando-se as duas menores notas** para o cômputo da média final. O aluno contará também com alguns **bônus**, em algumas atividades que serão indicados explicitamente pelos professores da disciplina.

Substitutiva

Não haverá.

Recuperação

Quem ficar com média final (MF) inferior a 5.5, poderá fazer uma avaliação de recuperação (REC) com duração de 72h que deverá ser entregue até o dia 15 de dezembro. Para quem decidir entregar a recuperação, a nota final do curso (NF) será obrigatoriamente $NF = (MF + REC) / 2$, ainda que este cálculo resulte numa nota inferior à MF. Para quem não entregar a prova de recuperação, a nota final do curso será $NF = MF$. Poderá fazer a recuperação, uma única vez, quem obtiver conceito final D ou F (ou seja, $MF < 5.5$).

Controle de frequência e reprovação por faltas

Conforme número de tarefas entregues.

Atividades e avaliações presenciais

Não haverá.

Conceito final

Nota final* (NF)	Conceito
$8.5 \leq NF$	A
$7.0 \leq NF < 8.5$	B
$5.5 \leq NF < 7.0$	C
$4.5 \leq NF < 5.5$	D
$NF < 4.5$	F

* Nenhuma nota será arredondada.

Programa por semanas

- I. Introdução
- II. Vetores e Matrizes
- III. Funções e Recursão
- IV. Recursão
- V. Ponteiros
- VI. Alocação dinâmica de memória
- VII. Registros (structs)
- VIII. Alocação dinâmica de matrizes
- IX. Listas Ligadas I
- X. Listas Ligadas II
- XI. Tópicos variados