

Campus: São José dos Campos/ICT		
Curso(s): Bacharelado em Ciência e Tecnologia		
Unidade Curricular (UC): Lógica de Programação		
Unidade Curricular (UC): <i>Introduction to Programming</i>		
Código da UC: 9394		
Docente Responsável/Departamento: Prof. Dr. Rodrigo Colnago Contreras / Departamento de Ciência e Tecnologia		Contato (e-mail): rodrigo.contreras@unesp.br
Docente (s) Colaborador/a (es/as)/ Departamento(s):		Contato (e-mail): [opcional]
Ano letivo: 2025/15	Termo: 1	Turno: Noturno
Nome do Grupo/Módulo/Eixo da UC: (se houver)		Idioma predominante em que a UC será oferecida: <input checked="" type="checkbox"/> Português <input type="checkbox"/> English <input type="checkbox"/> Español <input type="checkbox"/> Français <input type="checkbox"/> Libras <input type="checkbox"/> Outro:
UC: <input checked="" type="checkbox"/> Fixa <input type="checkbox"/> Eletiva <input type="checkbox"/> Optativa	Oferecida como: <input checked="" type="checkbox"/> Disciplina <input type="checkbox"/> Módulo <input type="checkbox"/> Estágio <input type="checkbox"/> Outro:	Oferta da UC: <input checked="" type="checkbox"/> Semestral <input type="checkbox"/> Anual
Ambiente Virtual de Aprendizagem: <input type="checkbox"/> Moodle <input type="checkbox"/> Classroom <input type="checkbox"/> Não se aplica <input checked="" type="checkbox"/> Outro: Discord		
Pré-Requisito (s) - Indicar Código e Nome (s) da (s) UC: Não há.		
Carga horária total (em horas): 72		
Carga horária teórica: 36	Carga horária prática: 36	Carga horária de extensão (se houver):
Se houver atividades de extensão, indicar código e nome do projeto ou programa vinculado na Pró-Reitoria de Extensão e Cultura (ProEC):		
Ementa: Introdução à computação; Noções de lógica; Conceitos e representação de algoritmos; Constantes e variáveis; Estruturas de controle; Vetores; Matrizes; Registros e uniões; Procedimentos, Funções com passagem de parâmetros por valor e referência; Recursividade; Introdução à linguagem de programação.		
Conteúdo programático: Parte 1. Introdução a computação; Introdução a lógica de programação; Noções de lógica; Algoritmos; Pseudocódigos e fluxogramas; Teste de mesa. Parte 2. Elementos básicos de algoritmos: Constantes, variáveis simples e compostas; Tipos enumerados; Comandos de entrada e saída; Expressões, estruturas sequenciais e condicionais; Estruturas de repetição; Funções. Parte 3. Linguagem de programação C (padrão ANSI): Sintaxe da linguagem; Modularização: procedimentos e funções (passagem de parâmetros por valor e referência); Funções recursivas. Vetores, matrizes, registros e uniões; Busca sequencial e binária em vetores;		
Objetivos: Gerais: Este curso busca desenvolver o raciocínio lógico dos alunos para a resolução de problemas computacionais de forma estruturada e eficiente, além de introduzir os conceitos fundamentais de programação e construção de algoritmos. Outro objetivo é capacitar os estudantes a projetar, implementar e depurar algoritmos e programas na linguagem de programação C, utilizando boas práticas de codificação. Específicos: Ao final do curso, os alunos deverão ser capazes de aplicar conceitos de lógica de programação para construir algoritmos eficientes e organizados, representando-os por meio de pseudocódigos e fluxogramas. Serão aptos a manipular dados utilizando variáveis, vetores, matrizes, registros e uniões, além de implementar estruturas de controle, como condições, loops e funções, para modularizar programas. Também estarão capacitados a desenvolver, analisar e depurar programas na linguagem C (padrão ANSI), aplicando boas práticas de codificação, utilizando funções recursivas e técnicas como busca sequencial e binária, e resolvendo problemas reais de forma estruturada e otimizada.		

Metodologia de ensino: As aulas serão ministradas por meio de exposições teóricas sobre os conceitos fundamentais, complementadas por atividades práticas de laboratório focadas na confecção de programas. Além disso, metodologias ativas de ensino, como a gamificação utilizando o Discord, serão eventualmente implementadas para tornar o aprendizado mais dinâmico e engajador. Quando disponíveis, os monitores irão auxiliar no desenvolvimento das atividades práticas solicitadas, proporcionando suporte adicional aos alunos durante o curso.

Avaliação: A avaliação se dará por meio de duas provas teóricas, P1 e P2, um trabalho prático, T, e determinadas atividades realizadas em sala que, somadas, são representadas pelo montante E, totalizando a nota final, NF, contabilizada como segue:

$$NF = \min\{(P1+P2+0.1*T + E) / 2, 10\},$$

na qual, P1, P2, T e E pertencem ao intervalo [0,10].

O aluno será considerado aprovado se NF pertencer ao intervalo [6,10] e reprovado no caso contrário.

Bibliografia:

Básica:

1. Forbellone, André LV; Eberspache, Henri F. Lógica de programação: a construção de algoritmos e estruturas de dados. 3.ed. São Paulo: Pearson, 2005. 218 p. ISBN 978-85- 7605-024-7.;
2. Feofiloff, Paulo. Algoritmos em linguagem C. Rio de Janeiro: Elsevier, 2009. 208 p. ISBN 978-85-352-3249-3.;
3. Mokarzel, Fábio; Soma, Nei. Introdução à ciência da computação. Rio de Janeiro: Elsevier, 2008. 429 p. ISBN 978-85-352-1879-4.;

Complementar:

1. Mizrahi, Victorine Viviane. Treinamento em linguagem C: módulo profissional. São Paulo: Makron, c1993. 225 p. ISBN 978-85-346-0109-2.;
2. Deitel, Paul; Deitel, Harvey. C: como programar. [C: how to program]. Tradução: Daniel Vieira. 6 ed. São Paulo: Pearson Prentice Hall, 2011. 818 p. ISBN 978-85-7605-934-0.;
3. KERNIGHAN, Brian W; VIEIRA, Daniel; RITCHIE, Dennis M. C: a linguagem de programação padrão ANSI. Rio de Janeiro: Campus, 1989. ISBN 978-85-7001-586-0.;
4. FARRER, Harry et al. Algoritmos estruturados. 3.ed. Rio de Janeiro: LTC, 1999. 284 p. ISBN 978-85-216-1180-6. ;
5. Horowitz, Ellis; Sahni, Sartaj; Rajasekaran, Sanguthevar. Computer algorithmics/C++. New York: Computer Science, 1997. 769 p. ISBN 978-0-7167-8315-2.