



Ficha 2 (variável)

(A modalidade das disciplinas ofertadas com base na Res. 65/2020 – CEPE, em respeito ao Parágrafo Único do Art. 1º desta resolução, deverá ser invariavelmente a modalidade de *ensino remoto emergencial* (ERE). Sendo assim, para essas disciplinas, fica dispensado o preenchimento do campo “Modalidade” desta Ficha 2 (Plano de Ensino), que não contempla essa modalidade de ensino.)

Disciplina: Programação 2						Código: CI1002	
Natureza: (x) Obrigatória () Optativa		(x) Semestral () Anual () Modular					
Pré-requisito:		Co-requisito:		Modalidade: () Presencial (x) Totalmente EaD () *C.H.EaD			
CH Total: 60 CH semanal: 8		Padrão (PD): 30	Laboratório (LB): 00	Campo (CP): 00	Estágio (ES): 00	Orientada (OR): 00	Prática Específica (PE): 30
Estágio de Formação Pedagógica (EFP):		Extensão (EXT): 00	Prática como Componente Curricular (PCC): 00				
<p>Indicar a carga horária semestral (em PD-LB-CP-ES-OR-PE-EFP-EXT-PCC) *Indicar a carga horária que será à distância.</p> <p style="text-align: center;">EMENTA (Unidade Didática)</p> <p>Uso de técnicas avançadas para desenvolvimento de software. Práticas de projetos de desenvolvimento de programas de média e alta complexidade.</p>							
PROGRAMA (itens de cada unidade didática)							
<ul style="list-style-type: none"> • Tratamento de strings • O préprocessador C • Conceitos avançados da linguagem C • Manipulação de arquivos • Aplicação avançada de ponteiros • Modularização de programas e Makefile • Técnicas e ferramentas de teste e depuração • Construção de Bibliotecas • Desenvolvimento de projetos de média e alta complexidade 							
OBJETIVO GERAL							
<p>Capacitar o estudante a desenvolver programas de média e alta complexidade usando técnicas de algoritmos e estruturas de dados já vistas em disciplinas anteriores, sempre considerando a noção de eficiência dos códigos desenvolvidos.</p>							
OBJETIVO ESPECÍFICO							
<ul style="list-style-type: none"> • Aprender a ler, escrever e manipular strings. • Compreender o funcionamento do préprocessador C e sua aplicação na construção de programas. • Conhecer aspectos avançados da linguagem: conversão de tipos (casting), argumentos da função main(), enumerações, uniões, bitfields. • Compreender a manipulação de arquivos em disco por um programa: entrada e saída padrão, leitura e gravação de dados em arquivos. • Estudar a aplicação avançada de ponteiros: matrizes dinâmicas, ponteiros de funções, uso de alocação durante leitura de arquivos. • Compreender o conceito de modularização de programas de média e alta complexidade e aprender a usar a ferramenta make. • Compreender o conceito e mecanismo de construção de bibliotecas em ambiente Linux. • Complementar os métodos de depuração e teste de programas iniciado em semestres anteriores, incluindo 							



- teste de módulos/funções e teste da caixa preta.
- Utilizar as técnicas e conceitos aprendidos no desenvolvimento de projetos de média e alta complexidade.

PROCEDIMENTOS DIDÁTICOS

O curso mesclará os seguintes procedimentos didáticos:

- Aulas expositivas assíncronas, por meio de vídeos gravados pelo professor e disponibilizados em plataforma a definir (AVA-UFPR, página pessoal do docente ou outro meio).
- Projetos e atividades relativos ao conteúdo apresentado.
- Encontros síncronos coletivos de 1h semanais, para discussão e esclarecimento de dúvidas, usando ferramenta a definir (Discord, Google Meet, Jitsi ou outro).
- Atendimento individualizado aos alunos sob demanda, por meio de e-mail e reuniões online.
- Controle de frequência pelas atividades síncronas e entrega dos projetos e atividades propostas.

O material didático da disciplina estará inteiramente disponível a partir da página pessoal do docente.

FORMAS DE AVALIAÇÃO

O desempenho dos alunos na disciplina será avaliado da seguinte forma:

- Dois projetos individuais desenvolvidos ao longo da disciplina e defendidos pelo aluno ao final.
- Exame final, cobrindo todo o conteúdo (teoria, projetos e atividades).
- Projetos, atividades e provas entregues fora dos seus respectivos prazos serão desconsiderados.

Critérios de aprovação:

- Nota final ≥ 70 : aprovação.
- Nota final < 40 : reprovação.
- Nota final ≥ 40 e < 70 : exame final, que deve ter nota ≥ 50 .

BIBLIOGRAFIA BÁSICA (mínimo 03 títulos)

- Brian W. Kernighan e Dennis M. Ritchie. C: A Linguagem de Programação Padrão ANSI. Ed Campus, 1990.
- Marco Medina e Cristina Fertig. Algoritmos e Programação: Teoria e Prática. 2 a . Ed. Novatec, 2006.
- A.M. Tenenbaum, Y. Langsam e M.J. Augenstein. Estruturas de Dados Usando C. Makron Books, 1995.

BIBLIOGRAFIA COMPLEMENTAR (mínimo 05 títulos)

- Nivio Ziviani. Projeto de algoritmos: com implementações em PASCAL e C. Ed. Pioneira Thomson, 2004.
- H. Schildt. C Completo e Total. 3 a . Makron Books, 1997.
- P. Prinz e U. Kirch-Prinz. C Pocket Reference. Ed. O'Reilly, 2003.
- J.L. Szwarcfiter e L. Markenzon. Estruturas de Dados e seus Algoritmos. LTC, 1994.
- Donald E. Knuth. The Art of Computer Programming. Addison-Wesley Publishing Company, 1997.

Obs: A bibliografia indicada deverá efetivamente estar disponível na biblioteca em número compatível com o tamanho de cada turma.

Professor da Disciplina: Lucas Ferrari de Oliveira

Assinatura: _____

Chefe de Departamento ou Unidade equivalente: Fabiano Silva

Assinatura: _____



Disciplina: CI-1002 – Programação 2

Vagas: 20

Docente responsável: prof. Lucas Ferrari de Oliveira (lferrari@inf.ufpr.br)

Carga horária: 60 horas, distribuídas em 7 semanas (média de 9 horas semanais)

Modalidades e meios:

- Atividades síncronas: videoconferência
- Atividades assíncronas: vídeos e textos em plataformas diversas

Cronograma detalhado (turma com início em 3/11/20)

Sem.	Data	Hora	Conteúdo teórico	Atividades práticas	Modalidade	Carga horária
1 de 20/09 a 24/09	21/09	15:30	Encontro semanal: Apresentação da disciplina	Apresentação Projeto 1	Síncrona	2
	-	-	Strings Codificação de caracteres Strings multibyte	Exercícios sobre o conteúdo Desenvolvimento do projeto	Assíncrona.	8
2 de 27/09 a 01/10	28/09	15:30	Encontro semanal		Síncrona	1
	-	-	Acesso a arquivos Arquivos binários	Exercícios sobre o conteúdo Desenvolvimento do projeto	Assíncrona	10
3 de 04/10 a 08/10	05/10	15:30	Encontro semanal		Síncrona	1
	-	-	Organização de código O preprocessador C O sistema Make	Exercícios sobre o conteúdo Desenvolvimento do projeto	Assíncrona	9
	10/10	23:59		Entrega projeto 1	Assíncrona	0
4 de 11/10 a 15/10	14/10	15:30	Encontro semanal	Apresentação Projeto 2	Síncrona	1
	-	-	Depuração A função main Tipos enumerados	Exercícios sobre o conteúdo Desenvolvimento do projeto	Assíncrona	8
5 de 18/10 a 22/10	19/10	15:30	Encontro semanal		Síncrona	2
	-	-	Conversão de tipos Uniãos Operações com bits Ponteiros para funções	Exercícios sobre o conteúdo Desenvolvimento do projeto	Assíncrona	8
6 de 25/10 a 29/10	26/10	15:30	Encontro semanal		Síncrona	1
	-	-	Bibliotecas Construção de bibliotecas	Exercícios sobre o conteúdo Desenvolvimento do projeto	Assíncrona	8
	03/11	23:59		Entrega projeto 2	Assíncrona	0
7 de 08/11 a 12/11	-	-		Defesas dos projetos 1 e 2 (entrevistas individuais)	Síncrona	1
8	23/11	15:30	Exame final, com livre consulta e defesa oral		Síncrona	-