



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE _____

Coordenação do Curso de ou Departamento
de _____

Ficha 2 (variável)

Disciplina: **Laboratório de Programação II** Código: **DEE342**

Natureza:
(x) Obrigatória (x) Semestral () Anual () Modular
() Optativa

Pré-requisito: Co-requisito: Modalidade: (x) Totalmente Presencial () Totalmente EAD () Parcialmente EAD: _____ *CH

CH Total: 60							
CH Semanal: 4							
Prática como Componente Curricular (PCC):	Padrão (PD):	Laboratório (LB): 60	Campo (CP):	Estágio (ES):	Orientada (OR):	Prática Específica (PE):	Estágio de Formação Pedagógica (EFP):
Atividade Curricular de Extensão (ACE):							

Indicar a carga horária semestral (em PD-LB-CP-ES-OR-PE-EFP-EXT-PCC)

*indicar a carga horária que será à distância.

EMENTA

Uso dirigido de ferramentas de desenvolvimento e depuração de software. Práticas de programação e projetos de algoritmos.

PROGRAMA

- 1. Conceitos avançados da linguagem C.**
- 2. Manipulação de arquivos.**
- 3. Aplicação avançada de ponteiros.**
- 4. Modularização de programas e Makefile.**
- 5. Construção de Bibliotecas.**

6. Métodos para teste de módulos/funções. Teste da caixa preta

7. Desenvolvimento de projetos de média e alta complexidade

Aula 1: Apresentação da disciplina, fichas 1 e 2

Aula 2: Relembrando o C

Aula 3: Prática de exercícios de revisão

Aula 4: Conceitos avançados da linguagem C

Aula 5: Conceitos avançados da linguagem C

Aula 6: Trabalho prático de implementação

Aula 7: Trabalho prático de implementação

Aula 8: Relembrando ponteiros

Aula 9: Aplicação avançada de ponteiros

Aula 10: Aplicação avançada de ponteiros

Aula 11: prática de ponteiros

Aula 12: Lista de exercícios

Aula 13: Modularização de programas

Aula 14: Modularização de programas e Makefile

Aula 15: Prática de modularização e makefile

Aula 16: Revisão

Aula 17: Prova escrita 1

Aula 18: Construção de Bibliotecas.

Aula 19: Construção de Bibliotecas.

Aula 20: criando uma biblioteca

Aula 21: Métodos para teste de módulos/funções. Teste da caixa preta

Aula 22: Métodos para teste de módulos/funções. Teste da caixa preta

Aula 23: Realizar testes de caixa preta

Aula 24: Desenvolvimento de projeto de média complexidade

Aula 25: Desenvolvimento de projeto de média complexidade

Aula 26: Defesa trabalho de média complexidade

Aula 27: Desenvolvimento de projetos de alta complexidade

Aula 28: Desenvolvimento de projeto de alta complexidade

Aula 29: Defesa projeto 1 de alta complexidade

Aula 30: Exame

OBJETIVO GERAL

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.

OBJETIVO ESPECÍFICO

1. Conceitos avançados da linguagem Conversão de tipos (casting), pré-processador C, argumentos da função main(), unions, bitfields.

2. Compreender a manipulação de arquivos em disco por um programa: entrada e saída padrão, leitura e gravação de dados em arquivos.
3. Estudar a aplicação avançada de ponteiros: matrizes dinâmicas, ponteiros de funções, uso de alocação durante leitura de arquivos.
4. Compreender o conceito de modularização de programas de média e alta complexidade, e sua relação com a ferramenta make.
5. Compreender o conceito e mecanismo de construção de bibliotecas em ambiente Linux.
6. Complementar os métodos de depuração e teste de programas iniciado em semestre anteriores, incluindo teste de módulos/funções e teste da caixa preta.
7. Utilizar as técnicas e conceitos aprendidos no desenvolvimento de projetos de média e alta complexidade.

PROCEDIMENTOS DIDÁTICOS

Aulas expositivas e práticas em laboratório com computadores e projetor, juntamente com desenvolvimento de projetos de programação em laboratório.

FORMAS DE AVALIAÇÃO

A avaliação será feita através de projetos realizados ao longo da disciplina. Serão realizados três ou mais projetos individuais, seguidos de demonstração e defesa dos projetos perante o professor da disciplina.

BIBLIOGRAFIA BÁSICA (mínimo 03 títulos)

MEDINA, M. A.; FERTIG, C. Algoritmos e Programação: Teoria e Prática. São Paulo: Novatec, 2005.

BARRY, P.; GRIFFITHS D. Use a Cabeça! Programação. Alta Books, 2010.

ZIVIANI, N. Projeto de Algoritmos. 3. ed. São Paulo: Cenage, 2010.

BIBLIOGRAFIA COMPLEMENTAR (mínimo 05 títulos)

ANICHE, M. Test-Driven Development: Teste e Design no Mundo Real. Casa do Código, 2012.

FELLEISEN, M.; FINDLER, R. B.; FLATT, M.; KRISHNAMURTHI, S. How to Design Programs. Cambridge: The MIT Press, 2003
Disponível em: <http://htdp.org/2003-09-26/>.

MARTIN, R. C. Código Limpo - Habilidades Práticas do Agile Software. Alta Books, 2009.

SILVEIRA, P.; ALMEIDA A. Lógica de Programação. Casa do Código, 2013.



Documento assinado eletronicamente por **JEFER BENEDETT DORR, PROFESSOR DO MAGISTERIO SUPERIOR**, em 29/11/2021, às 15:19, conforme art. 1º, III, "b", da Lei 11.419/2006.



A autenticidade do documento pode ser conferida [aqui](#) informando o código verificador **4073414** e o código CRC **3193C74B**.
