



## PROGRAMA GERAL DO COMPONENTE CURRICULAR- PGCC<sup>1</sup>

### I IDENTIFICAÇÃO DO COMPONENTE CURRICULAR

**1.1 Natureza do componente:** (X)Disciplina ( )Atividades da prática<sup>2</sup> ( )Estágio Supervisionado Obrigatório

( )Trabalho de Conclusão de Curso – TCC

**1.2 Nome do componente:** Construção de Algoritmos

**CÓDIGO:** 0805016-1

**CRÉDITOS:** 06

**CARGA HORÁRIA:** 90 h.a.

**Pré-Requisito:** Lógica Matemática Aplicada à Computação

**Código:** 0805034-1

**Curso:** Ciência da Computação

**Período:** 2º

**Turno:** Matutino

**Ano/Semestre:** 2012.2

**Professor (a):** Sebastião Emídio Alves Filho

### II EMENTA

Divisão da lógica computacional; Conceitos de programação estruturada; Algoritmos; Tipos de dados; Estrutura de controle; Estrutura de dados homogêneas; Classificação e pesquisa; Estruturas de dados heterogêneas; Sub-algoritmos e recursividade; Arquivos.

### III OBJETIVOS

Divisão da Lógica Computacional. Conceitos de programação Estruturada. Algoritmos. Tipos de Dados.

Estrutura de Controle. Estrutura de dados homogêneas. Classificação e pesquisa. Estruturas de dados

heterogêneas. Sub-algoritmos e recursividade. Arquivos. Introduzir os conceitos de resolução de problemas através do computador usando algoritmos estruturados, fazendo com que o aluno seja capaz, ao término da disciplina, de modelar e implementar soluções através de algoritmos e implementá-los na linguagem de programação estruturada C.

### IV CONTEÚDO

#### 1a Unidade

#### 1. Conceitos básicos

1.1. Conceito de algoritmo e fluxo de execução de programas

1.2. Formas de representação

1.3. Técnicas de resolução de Problemas usando CS-Unplugged

1.4. Técnicas de resolução de Problemas usando Scratch (Laboratório)

#### 2. Tipos de dados e sua representação computacional

2.1. Dados numéricos - Inteiro e ponto flutuante; dados Caracteres - Caractere e String; dados lógicos – Booleano

2.2. Conceito de endereçamento e ponteiros

2.3. Dados compostos homogêneos (enumerações, vetores e matrizes) e heterogêneos (estruturas);

2.4. Construção de algoritmos em pseudo-código: Variáveis e constantes – Declaração, inicialização e leitura/escrita

2.5. Introdução à linguagem C: edição, compilação e execução de programas(Laboratório)

2.6. Introdução à linguagem C: criação de variáveis, entrada e saída padrão (Laboratório)

#### 3. Operações com variáveis

3.1. Operações aritméticas – soma, subtração, multiplicação, divisão inteira e real, resto

3.2. Avaliação de expressões aritméticas - Precedência de operadores e uso dos parênteses

- 3.3. Operações de comparação -  $>$ ,  $<$ ,  $\leq$ ,  $\geq$ ,  $=$ ,  $\neq$
- 3.4. Operações lógicas - e, ou, não, ou-exclusivo
- 3.5. Conversão entre tipos diferentes
- 3.6. Operações com variáveis e precedência de operadores em C (Laboratório)

#### **4. Desvios condicionais e incondicionais**

- 4.1. Operador “se ... então”
- 4.2. Operador “se ... então ... senão ...”
- 4.3. Condicionais aninhados
- 4.4. Operador “caso”
- 4.5. Operadores condicionais if, if/else e switch/case do C (Laboratório)
- 4.6. Conceito de rótulo e comando “vá para” (goto)

### **2a Unidade**

#### **5. Estruturas de repetição**

- 5.1. Laço com número definido de passos: “Para”
- 5.2. Laço com número indefinido de passos e teste no início: “Enquanto”
- 5.3. Laço com número indefinido de passos e teste no fim: “Repita até que”
- 5.4. Laços aninhados; Modificações no fluxo de um laço
- 5.5. Laços “for”, “while”, “do/while” em C (Laboratório)
- 5.6. Comandos break e continue em C (Laboratório)

#### **6. Estruturas de dados homogêneas**

- 6.1. Exemplos de algoritmos que utilizam vetores e matrizes
- 6.2. Manipulação de cadeias de caracteres
- 6.3. Métodos simples de ordenação de valores em um vetor
- 6.4. Métodos simples de busca em um vetor
- 6.5. Manipulação de vetores em C (Laboratório)
- 6.6. Manipulação de strings em C (Laboratório)

### **3a Unidade**

#### **7. Estruturas de dados heterogêneas e arquivos**

- 7.1. Exemplos de algoritmos que utilizam estruturas (registros)
- 7.2. Vetores e matrizes de registros
- 7.3. Tipos de arquivos e operações simples com arquivos: abertura, fechamento, leitura e escrita
- 7.4. Operações com os tipos “struct” definidos em C (Laboratório)
- 7.5. Manipulação de arquivos em C (Laboratório)

#### **8. Modularização de programas**

- 8.1. Vantagens da divisão de programas em módulos
- 8.2. Tipos de subprogramas: funções e procedimentos
- 8.3. Conceito de variáveis locais e globais - Escopo de variáveis
- 8.4. Tipos de passagens de parâmetros: por valor e por referência
- 8.5. Conceito de recursividade
- 8.6. Funções em C (Laboratório)
- 8.8. Passagem por referência em C (Laboratório)

## **V METODOLOGIA**

- Aulas expositivas em sala
- Aulas práticas em laboratório com o software Scratch
- Aulas práticas em laboratório de programação com os IDEs Anjuta e Dev-Cpp
- Utilização do AVA Moodle
- Utilização do portal URI Onlinejudge

## VI PROCEDIMENTOS DE AVALIAÇÃO DA APRENDIZAGEM

- 1ª unidade – Prova escrita, listas de exercícios práticos em Scratch, pseudo-código e em C
- 2ª unidade – Prova escrita, lista de exercício prático em C, seminário sobre as funções da biblioteca C Padrão
- 3ª unidade – Prova escrita, projeto prático final da disciplina

## VII REFERÊNCIAS

### Bibliografia básica

MANZANO, José Augusto N. G.; OLIVEIRA, Jayr Figueiredo. Algoritmos: lógica para desenvolvimento de programação. Ed. Érica, 2000.

MIZRANI, Victorine Viviane. Treinamento em Linguagem C, Módulo 1. Editora Makron Books, 1990.

OLIVEIRA, Ulysses de. Programando em C - Volume I - Fundamentos, Editora Ciência Moderna, 2008.

### Bibliografia Complementar

DEITEL, H. M. Java: Como Programar, Prentice Hall, 2006.

MIZRANI, Victorine Viviane. Treinamento em Linguagem C, Módulo 2. Editora Makron Books, 2001.

MIZRANI, Victorine Viviane. Treinamento em Linguagem C++, Módulo 1. Editora Makron Books, 1994.

SALVETTI, Dirceu D.; MADSEN, Barbosa Lisbete. Algoritmo. Ed. Makron Books, 2004.

ZIVIANI, Nivio. Projeto de Algoritmos : com Implementações em Pascal e C , Pioneira, 1999.

### Ferramentas e sites de apoio

Scratch – <http://scratch.mit.edu/>

Anjuta – [projects.gnome.org/anjuta/](http://projects.gnome.org/anjuta/)

Dev-Cpp – [www.bloodshed.net/devcpp.html](http://www.bloodshed.net/devcpp.html)

CS Unplugged – [csunplugged.org/](http://csunplugged.org/)

Computer Science for Fun – [www.cs4fn.org/magic/](http://www.cs4fn.org/magic/)

URI Online Judge – [urionlinejudge.com.br/](http://urionlinejudge.com.br/)

Site da Olimpíada Brasileira de Informática - [olimpiada.ic.unicamp.br/](http://olimpiada.ic.unicamp.br/)

Moodle–DI – [di.uern.br/ead](http://di.uern.br/ead)

## VIII OUTRAS OBSERVAÇÕES

Aprovado pela Comissão do PPC em \_\_\_\_/\_\_\_\_/\_\_\_\_

\_\_\_\_\_  
Professor(a)

\_\_\_\_\_  
Presidente da Comissão do PPC