



# ESTRUTURA DE DADOS E ALGORITMOS I

Apresentação da Disciplina

Profº. Sérgio Roberto Costa Vieira, M.Sc.

Cursos de Computação

2º. Período

# Apresentação da Disciplina

## Professor

- **Sérgio Roberto Costa Vieira**
  - Colaborador da FUCAPI há 14 anos.
  - Docente desde 2012.
- **Formação:**
  - Mestrado em Informática pela Universidade Federal do Amazonas (UFAM).
  - Graduado Bacharel em Analise de Sistemas pela Faculdade FUCAPI.
- **Atuação:**
  - Atualmente é Professor efetivo da Faculdade FUCAPI e,
  - Membro do grupo de pesquisa Usabilidade e Engenharia de Software (USES/UFAM).
  - Linha de Pesquisa: Engenharia de Software → Requisitos

# Apresentação da Disciplina

## Dados Gerais

- **Dados da Disciplina**
  - Nome: **Estrutura de Dados e Algoritmos I**
  - Carga Horária: **80h**
  - Período: **2º. / 2016-2**
  - Horário das Aulas:



### Tarde

INF02VA: Segunda e Quarta – 1º. Horário

### Noite

INF02NA: Terça e Quinta – 2º. Horário

# Apresentação da Disciplina

## Objetivo



- **Geral**

- Capacitar e habilitar o aluno na compreensão de **como são elaboradas as estruturas de dados mais comuns utilizadas em programas de computador para resolução de problemas do mundo real, implementado essas estruturas através de Tipos Abstratos de Dados (TAD) em Linguagens de Programação.**

# Apresentação da Disciplina

## Objetivos



- **Específicos**

- Possibilitar a resolução de **problemas complexos utilizando TAD's;**
- Apresentar as várias formas de estruturação de dados **em memória e em arquivo;**
- Estudar e implementar os algoritmos que manipulam, organizam e ordenam dados **utilizando como base uma estrutura de dados.**



# Apresentação da Disciplina

## Ementa

- **Introdução:** *Algoritmos. Estrutura de Dados e Programas. Tipos de Dados e Tipos Abstratos de Dados. Técnicas de Linguagem de Programação.*
- **Tipos de Dados Primitivos:** *cadeia de caracteres, apontadores (ponteiros).*
- **Estruturas de Dados Básicas:** *Variáveis compostas homogêneas unidimensionais (vetores) e multidimensionais (matrizes). Variáveis compostas heterogêneas (registros ou estruturas) e conjuntos de registros.*
- **Modularização:** *Procedimentos e funções iterativas e recursivas.*
- **Arquivos:** *conceitos, organização, abertura, fechamento, organização sequencial (entrada e saída), acesso e atualização à Registro de Arquivo.*

# Apresentação da Disciplina

## Conteúdo Programático

- **Introdução**
  - Algoritmos
  - Estrutura de Dados e Programas
  - Tipos de Dados e Tipos Abstratos de Dados
  - Técnicas de Linguagem de Programação
  - Tipos de Dados Primitivos: cadeia de caracteres
- **Estruturas de Dados Básicas**
  - Variáveis compostas homogêneas unidimensionais (vetores)
  - Variáveis compostas homogêneas Multidimensionais (matrizes)
  - Manipulação de cadeia de caracteres (strings)
- **Modularização**
  - Definição de funções e procedimentos
  - Parâmetros de funções
  - Passagem de parâmetros por valor
  - Passagem de parâmetros por referência
- **Estruturas de Dados Avançadas**
  - Operações com Recursão
  - Variáveis compostas heterogêneas (registros ou estruturas)
  - Modularização com registros
- **Ponteiros**
  - Definição
  - Operações com ponteiros
- **Arquivos**
  - Conceitos
  - Organização
  - Abertura
  - Fechamento
  - Organização sequencial (entrada e saída)
  - Acesso e atualização à Registro de Arquivo

# Apresentação da Disciplina

## Metodologia de Ensino

- **Metodologia**
  - Aulas expositivas
  - Trabalhos em grupos
  - Listas de Exercícios
  - Atividades práticas em Laboratórios



- **Recursos**
  - Sala de Aula, Quadro, Data-show, Pincéis...



# Apresentação da Disciplina

## Avaliação

- A média semestral será obtida através da fórmula:

$$MS = \left[ \frac{(1^a NB) + (2^a NB)}{2} \right]$$

- 1ª Nota Bimestral:  $1^a NB = \left[ \frac{(AP_1) + (AB_1)}{2} \right]$ 
  - AP - Avaliação Parcial [10 pts] – 02 Provas valendo 5,0 pts
- AB<sub>1</sub> - Avaliação Bimestral [10 pts]
  - Aplicada na Semana de Prova – 02 Provas valendo 5,0 pts

# Apresentação da Disciplina

## Avaliação

- A média semestral será obtida através da fórmula:

$$MS = \left[ \frac{(1^a NB) + (2^a NB)}{2} \right]$$

- 2ª Nota Bimestral:  $2^a NB = \left[ \frac{(AP_2) + (AB_2)}{2} \right]$ 
  - AP - Avaliação Parcial [10 pts] – 02 Provas valendo 5,0 pts
- AB<sub>2</sub> - Avaliação Bimestral [10 pts]
  - Aplicada na Semana de Prova – 02 Provas valendo 5,0 pts

# Apresentação da Disciplina

## Avaliação

- Se a nota da média semestral for:
  - inferior a 7,5 e igual ou superior a 3,0:
    - o aluno deverá realizar o Exame Final (EF), que abrangerá todo o conteúdo ministrado no semestre, devendo ser obrigatoriamente individual.
  - superior ou igual a 7,5:
    - o aluno estará dispensado de realizar o Exame Final (EF).
- Assim, a média final do aluno será calculada através da seguinte fórmula:
$$MSE = \left[ \frac{(EF) + (MS)}{2} \right]$$
  - O aluno somente estará aprovado
    - se obtiver nota igual ou superior a 5,0 (cinco) e,
    - frequência mínima de 75% (setenta e cinco por cento).

# Apresentação da Disciplina

## Avaliação

- O aluno que não se submeter às avaliações bimestrais e parciais:
  - poderá fazer a prova de segunda chamada
    - desde que requeira na Secretaria Acadêmica obedecendo a data fixada no Calendário Acadêmico.
- Justificativas:
  - Atestado médico original
  - Convocação militar
  - Prorrogação da jornada de trabalho mediante declaração
  - Convocação pela Justiça Comum, Trabalhista ou Eleitoral

# Apresentação da Disciplina

## Bibliografia



- **Básica**

- ZIVIANI, N. Projeto de Algoritmos com Implementações em Pascal e C. 2. ed. São Paulo: Pioneira, 2004.
- GOODRICH, Michael, T TAMASSIA, Robert. Estrutura de Dados e Algoritmos em Java. 4. ed. Porto Alegre: Editora Bookman, 2007.
- DROZDEK, Adan. Estrutura de Dados e Algoritmos em C++. São Paulo: Pioneira Thomson, 2005.

# Apresentação da Disciplina

## Bibliografia

- **Complementar**



- **TENEMBAUN. Aarom M. Estrutura de Dados usando C. São Paulo: Makron Books, 1995.**
- **WIRTH, N. Algoritmos e Estrutura de Dados. Rio de Janeiro: PHB, 1989.**
- **SCHILDT, Herbert. C Completo e Total. 3. ed. São Paulo: Makron Books, 1997.**
- **CORMEN, Herbert. Algoritmos – Teorias e Práticas. 2. ed. Rio de Janeiro: Elsevier, 2002.**



# Apresentação da Disciplina

## Regras para uma Boa Relação

- Assistir às aulas com atenção, assiduidade e pontualidade;
- Os horários das aulas devem ser cumpridos
  - o aluno só é APROVADO se obter o mínimo de 75% de frequência na disciplina.
  - evitar o máximo de atrasos para não atrapalhar a aula.
- Justificativas....



# Apresentação da Disciplina

## Regras para uma Boa Relação

- **Manter os celulares em modo silencioso.**
  - caso precise atender o celular, isto deve ser feito fora da sala.
- **Alimentos deverão ser consumidos na cantina ou fora do horário de aula.**
- **Trabalhos de pesquisa devem seguir a formatação acadêmica da ABNT.**
  - Tomar cuidado com cópias de textos (plágio).
  - Indicar corretamente as fontes bibliográficas consultadas.

# Apresentação da Disciplina

## Calendário Acadêmico – 2º. Semestre

AGOSTO						
D	S	T	Q	Q	S	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

8	Início das Aulas para os Calouros. (Início do 1º Bimestre)
8	Início das Aulas para os Veteranos (Início do 1º Bimestre)
15	Divulgação do edital de monitoria
22	Data limite para aproveitamento de estudos para 2016/2
23	Início de período para aproveitamento de estudos para 2017/1
24 a 31	Período para Ajuste de Matrícula.
6, 13, 20, 27	Sábados Letivos

SETEMBRO						
D	S	T	Q	Q	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	

5	Feriado Estadual - Elevação do Amazonas à Categoria de Província
6	Dia Livre (recesso concedido)
7	Feriado Nacional - Independência do Brasil
23 a 30	Semana de provas 1º Bimestre
13 e 14	Evento Administração
3, 10, 17, 24	Sábados Letivos

OUTUBRO						
D	S	T	Q	Q	S	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

23/set a 03/out	Requerimento de Avaliação de 2ª chamada/1º Bimestre.
7	Fim do 1º Bimestre - limite para aplicação de provas
10	Início do 2º Bimestre
10	Data limite para lançamento da 1ª Bimestral/Faltas.
11	Divulgação dos Resultados da 1ª Bimestral/Faltas
12	Feriado Nacional - Nossa Senhora Aparecida
13	Data limite da divulgação da 2ª chamada.
15	Dia Livre - Dia dos Professores
14 a 17	Período para requerimento de revisão de notas(1º Bimestral)
24	Feriado Municipal - Aniversário de Manaus
22	Reunião Pedagógica
27 e 28	IV Congresso de Design
31	Exame de Proficiência
01 a 31	Outubro Rosa - NURE
1, 8, 22, 29	Sábados Letivos

# Apresentação da Disciplina

## Calendário Acadêmico – 2º. Semestre

NOVEMBRO						
D	S	T	Q	Q	S	S
		1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30			

2
9 a 11
14
15
17 e 18
28
29
30
1 a 30
5, 12, 19, 26

Feriado Nacional - Dia de Finados  
III ENCOENG  
Data Limite para Trancamento Parcial ou Integral  
Feriado - Proclamação da República  
Semana Global do Empreendedorismo  
Data limite para requerimento de Avaliação de 2ª chamada/2º Bimestral.  
Data limite para lançamento da 2ª Bimestral/ Faltas  
Divulgação dos Resultados da 2ª Bimestral/ Faltas  
Torneio Esportivo (masculino)  
Sábados Letivos

DEZEMBRO						
D	S	T	Q	Q	S	S
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

1 a 9
1 e 12
8
14
14
14 a 15
14 a 16
16
16
16
16
19 a 21
23
23
3, 10, 17
3 a 4/jan/2017
06/jan/2017

Semana de Provas (2º Bimestre)  
Requerimento de Avaliação 2ª Chamada (2º Bimestral)  
Feriado - Nossa Senhora da Conceição  
Data limite para lançamento de 2ª Bimestral/Faltas  
Divulgação dos Resultados da 2ª Bimestral/Faltas  
Período para Requerimento de Revisão de Notas (2º Bimestral)  
Aplicação de prova de segunda chamada 2º Bimestre  
Divulgação dos Resultados dos Requerimentos de Revisão (2º Bimestral)  
Limite para Lançamento de Notas das Avaliações de 2ª chamada (2º Bimestral)  
Divulgação de Listagem de Alunos em Exame Final  
Fim do 2º Bimestre  
Exames Finais  
Limite para lançamento dos Exames Finais  
Divulgação dos Resultados dos Exames Finais  
Sábados Letivos  
Período de Requerimento de Revisão de Exames Finais  
Resultados dos Requerimentos de Revisão dos Exames Finais

# Apresentação da Disciplina

## Contatos

- **Sérgio Roberto C. Vieira**
  - E-mail: [sergio.rcvieira@gmail.com](mailto:sergio.rcvieira@gmail.com)



- **Horário de Atendimento:**  
**Terça, Quinta e Sexta – Tarde – 15h às 18h**

**Deve ser agendado com antecedência por e-mail**

**Sala: F1 – Bloco F**

# Apresentação da Disciplina

## Fale sobre você...

- Nome
- Trabalha?
  - Onde?
  - Qual função?
  - Horário?
- Senão trabalha:
  - O que ocupa seu tempo?
- Qual o conteúdo de Introdução a Programação que você teve mais dificuldade?





# Revisão de Estruturas de Repetição

## Conceitos

### LAÇO CONDICIONAL COM TESTE NO INÍCIO

Caso seja necessário executar mais de uma instrução para uma condição verdadeira dentro de um laço, elas devem estar em um bloco com símbolos de chaves.

A instrução **while** deve ser escrita:

**while ( condição )**

*Lembrar que a variável que é testada na condição deve possuir um valor inicial antes do teste da condição.*

*Tomar cuidado  
com Laços  
infinitos*

**instruções...**

*Dentro do laço, deve atualizar esse valor por meio de incremento ou decremento.*

### LAÇO CONDICIONAL COM TESTE NO FINAL

Uma característica deste laço é que **a estrutura faz um teste lógico no final**, permitindo que a ação subordinada ao laço **seja executada no mínimo uma vez**.

A instrução **do...while** deve ser escrita:

**do**

**{**

*instruções...*

**} while( condição );** **→ Tomar cuidado com Laços infinitos**

*A variável que é testada na condição pode receber um valor inicial antes do teste da condição, dentro do próprio laço.*

### LAÇO CONTADO

Esse laço de repetição tem o funcionamento controlado por uma variável contadora, que pode ser crescente ou decrescente, tendo como sintaxe:

A instrução **for** deve ser escrita:

```
for ( início ; fim <condição> ; incremento )  
{  
    instruções...  
}
```

### LAÇO CONTADO

```
for ( início ; fim <condição> ; incremento )  
{  
    instruções...  
}
```

Em que:

**início** – é uma instrução de atribuição com o valor inicial do laço.

**fim** – é uma instrução de condição com o valor final do laço.

**incremento** – expressão com o incremento do laço.

### LAÇO CONDICIONAL COM TESTE NO INICIO

*Criar um programa que implemente a quantidade de termos que deve ser gerada pela série de FIBONACCI. A série é formada pela sequência: 1, 1, 2, 3, 5, 8, 13, 21, ...*

### LAÇO CONDICIONAL COM TESTE NO FINAL

*Criar um programa que leia vários números inteiros positivos e identificar o maior, o menor e média de todos os números informados. O número 0 (zero) é o flag de saída:*

### LAÇO CONTADO

*Criar um programa que gere a quantidade de termos da série de FETUCCINE. A série é formada pelas regras:*

- *O usuário informa os dois primeiros termos*
- *A partir daí, os próximos são gerados com base nas fórmulas:*

$$A_i = A_{i-1} + A_{i-2} \text{ (quando o } i \text{ for o valor ímpar)}$$

$$A_i = A_{i-1} - A_{i-2} \text{ (quando o } i \text{ for o valor par)}$$

*\*O  $i$  representa a posição do número*



# Revisão de Estruturas de Repetição

## Aninhamento ou Encadeamento

**Semelhantes às estruturas de decisão composta, as estruturas também podem ser encadeadas ou aninhadas.**

**Isto significa que, é possível colocar uma estrutura de repetição dentro da outra, ou combinar as estruturas de repetição diferentes.**

**Essa abordagem é utilizada quando há necessidade de usar laços dentro de laços.**

# Revisão de Estruturas de Repetição

## Aninhamento ou Encadeamento

```
1  #include<stdio.h>
2  #include<conio.h>
3
4  int main(){
5      /* Aninhamento de Repetições
6      Trata-se do uso de um comando de repetição
7      (while, for ou do..while) dentro do outro*/
8      //Exemplo
9      repetição(condição 1) {
10         sequência de comandos;
11         repetição(condição 2) {
12             sequência de comandos;
13         }
14     }
15     getch();
16     return 0;
17 }
18
19
20
```

### LAÇO CONDICIONAL ANINHADO

#### Teste no Inicio

```
while ( <condição> ){  
    while ( <condição> ){  
        instruções...  
    }  
}
```

#### Teste no Final

```
do{  
    do{  
        instruções...  
    } while ( <condição> )  
} while ( <condição> )
```

### LAÇO CONTADO ANINHADO

```
for ( inicio ; fim <condição> ; incremento ) {
```

```
    for ( inicio ; fim <condição> ; incremento ) {
```

```
        instruções...
```

```
    }
```

```
}
```

# Revisão de Estruturas de Repetição

## Aninhamento ou Encadeamento

### Exemplo com for

```
1  #include<stdio.h>
2  #include<conio.h>
3
4  int main()
5  {
6      int i, j;
7      for(i=1; i<5; i++){
8          for(j=1; j<5; j++){
9              if(i==j)
10                 printf("1 ");
11             else
12                 printf("0 ");
13         }
14         printf("\n");
15     }
16     getch();
17     return 0;
18 }
19
20
```

# Revisão de Estruturas de Repetição

## Aninhamento ou Encadeamento

### Exemplo com while

```
1  #include<stdio.h>
2  #include<conio.h>
3
4  int main()
5  {
6      int i=1, j;
7      while(i<5){
8          j = 1;
9          while(j<5){
10             if(i==j)
11                 printf("1 ");
12             else
13                 printf("0 ");
14             j++;
15         }
16         printf("\n");
17         i++;
18     }
19     getch();
20     return 0;
21 }
22
23
```



# Revisão de Estruturas de Repetição

## Aninhamento ou Encadeamento

**Exemplo com while  
combinado com um for**

```
1  #include<stdio.h>
2  #include<conio.h>
3
4  int main()
5  {
6      int i=1, j;
7      while(i<5){
8          for(j=1; j<5; j++){
9              if(i==j)
10                 printf("1 ");
11             else
12                 printf("0 ");
13         }
14         printf("\n");
15         i++;
16     }
17     getch();
18     return 0;
19 }
20
```

Conforme já foi estudado em estrutura de decisão ou condicional, o comando **break** pode ser utilizado em conjunto com o comando **switch**.

```
switch ( <expressão> ){  
    case 1: instruções; break;  
    case 2: instruções; break;  
    case 3: instruções; break;  
    default: instruções;  
}  
}
```

Na verdade o comando **break** serve para quebrar a execução de um conjunto de comandos (**como no caso do switch**) ou interromper a execução de qualquer laço de repetição (**while, for ou do..while**).

O comando **break** faz com que a execução do programa continue na primeira linha seguinte ao laço que está sendo interrompido.

# Revisão de Estruturas de Repetição

## Comando break

### Exemplo do comando break

```
1  #include<stdio.h>
2  #include<conio.h>
3
4  int main()
5  {
6      int a,b;
7      printf("Digite o valor de a: ");
8      scanf("%d", &a);
9      printf("Digite o valor de b: ");
10     scanf("%d", &b);
11     while (a <= b){
12         a = a + 1;
13         if(a == 5)
14             break;
15         printf("%d \n", a);
16     }
17     getch();
18     return 0;
19 }
20
```

# Revisão de Estruturas de Repetição

## Comando continue

O comando **break** quebra a execução do laço de repetição que está sendo executado.

O comando **continue** interrompe apenas aquela repetição e passa para a próxima repetição do laço, se ela existir.

Por esse motivo, o comando **continue** só pode ser usado dentro de um laço de repetição, diferentemente do comando **break**.

# Revisão de Estruturas de Repetição

## Comando continue

### Exemplo do comando continue

```
1  #include<stdio.h>
2  #include<conio.h>
3
4  int main()
5  {
6      int a,b;
7      printf("Digite o valor de a: ");
8      scanf("%d", &a);
9      printf("Digite o valor de b: ");
10     scanf("%d", &b);
11     while (a <= b){
12         a = a + 1;
13         if(a == 5)
14             continue;
15         printf("%d \n", a);
16     }
17     getch();
18     return 0;
19 }
20
```

### LAÇO CONDICIONAL ANINHADO (WHILE)

*Criar um programa que implemente a forma de um triângulo retângulo com N linhas, usando o símbolo de asterisco. Utilize repetição aninhada:*

```
*  
* *  
* * *  
* * * *  
* * * * *  
* * * * * *  
* * * * * * *  
* * * * * * *
```



### LAÇO CONDICIONAL ANINHADO (DO..WHILE)

*Criar um programa que leia um número  $N$  que indica quantos valores inteiros e positivos devem ser lidos a seguir. Para cada número lido, apresente o fatorial desse número:*

*Exemplo:*

*$Lê (número) = 5$*

*fatorial  $5! = 120$*

*$\rightarrow 5 \times 4 \times 3 \times 2 \times 1$*

# Revisão de Estruturas de Repetição

## Exercícios de Fixação

### LAÇO CONTADO ANINHADO (FOR)

*Criar um programa que implemente a tabuada completa de uma operação matemática do 2 ao 9, semelhante ao exemplo da figura abaixo:*

Tabuada do 2	Tabuada do 3	Tabuada do 4	Tabuada do 5
2 x 1 = 2	3 x 1 = 3	4 x 1 = 4	5 x 1 = 5
2 x 2 = 4	3 x 2 = 6	4 x 2 = 8	5 x 2 = 10
2 x 3 = 6	3 x 3 = 9	4 x 3 = 12	5 x 3 = 15
2 x 4 = 8	3 x 4 = 12	4 x 4 = 16	5 x 4 = 20
2 x 5 = 10	3 x 5 = 15	4 x 5 = 20	5 x 5 = 25
2 x 6 = 12	3 x 6 = 18	4 x 6 = 24	5 x 6 = 30
2 x 7 = 14	3 x 7 = 21	4 x 7 = 28	5 x 7 = 35
2 x 8 = 16	3 x 8 = 24	4 x 8 = 32	5 x 8 = 40
2 x 9 = 18	3 x 9 = 27	4 x 9 = 36	5 x 9 = 45
2 x 10 = 20	3 x 10 = 30	4 x 10 = 40	5 x 10 = 50
Tabuada do 6	Tabuada do 7	Tabuada do 8	Tabuada do 9
6 x 1 = 6	7 x 1 = 7	8 x 1 = 8	9 x 1 = 9
6 x 2 = 12	7 x 2 = 14	8 x 2 = 16	9 x 2 = 18
6 x 3 = 18	7 x 3 = 21	8 x 3 = 24	9 x 3 = 27
6 x 4 = 24	7 x 4 = 28	8 x 4 = 32	9 x 4 = 36
6 x 5 = 30	7 x 5 = 35	8 x 5 = 40	9 x 5 = 45
6 x 6 = 36	7 x 6 = 42	8 x 6 = 48	9 x 6 = 54
6 x 7 = 42	7 x 7 = 49	8 x 7 = 56	9 x 7 = 63
6 x 8 = 48	7 x 8 = 56	8 x 8 = 64	9 x 8 = 72
6 x 9 = 54	7 x 9 = 63	8 x 9 = 72	9 x 9 = 81
6 x 10 = 60	7 x 10 = 70	8 x 10 = 80	9 x 10 = 90



# ESTRUTURA DE DADOS E ALGORITMOS I

Apresentação da Disciplina

Profº. Sérgio Roberto Costa Vieira, M.Sc.

Cursos de Computação

2º. Período