



ESTRUTURA DE DADOS

LAÇOS E VETORES

Prof. Dr. Daniel Caetano

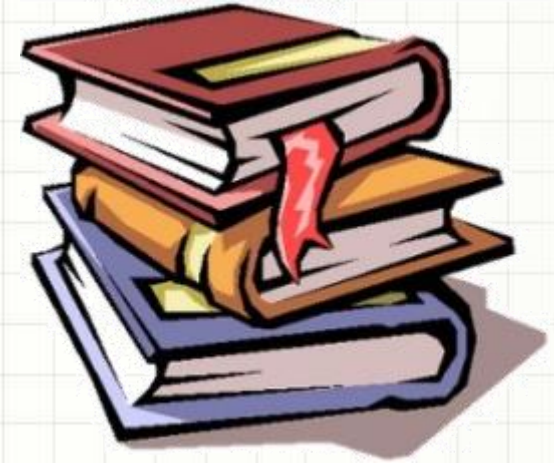
2014 - 2

Objetivos

- Recordar os laços de repetição
- Conhecer os vetores
- Compreender o uso de vetores



Material de Estudo



Material

Acesso ao Material

Notas de Aula

<http://www.caetano.eng.br/>
(Aula 2)

Apresentação

<http://www.caetano.eng.br/>
(Aula 2)

Material Didático

Estruturas de Dados, capítulo Listas Lineares



MOMENTO LÚDICO:
REPETIÇÃO

Momento Lúdico

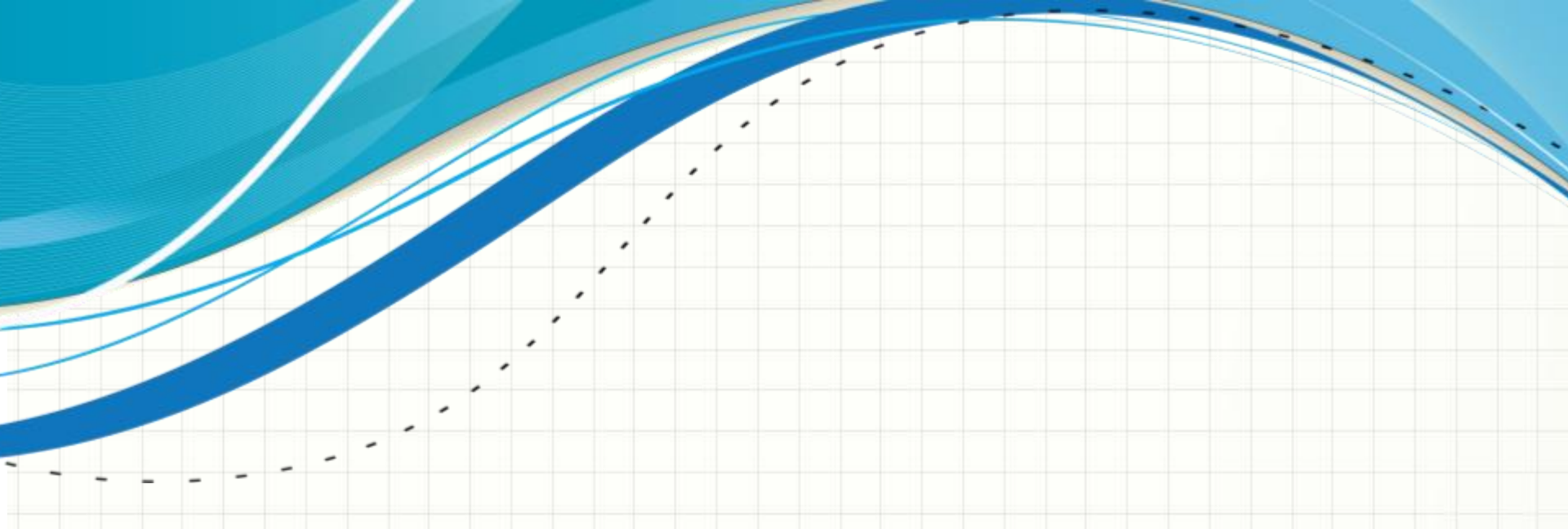
- É legal ficar repetindo uma tarefa?



Momento Lúdico

- Em geral... o computador não se importa!





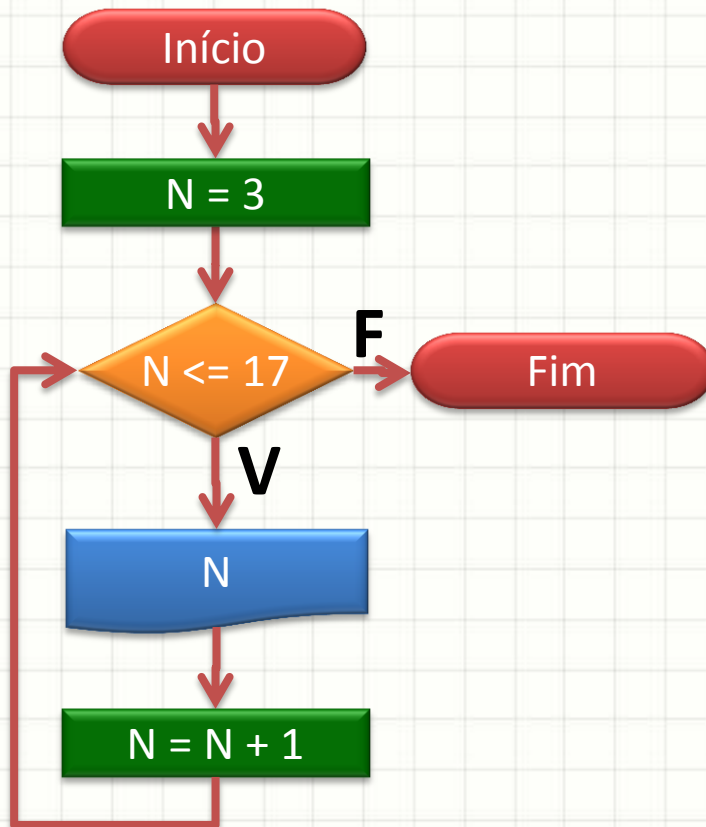
RECORDANDO:

REPETIÇÃO COM WHILE

Recordando o While

- Estrutura de repetição **while**

- O que faz?



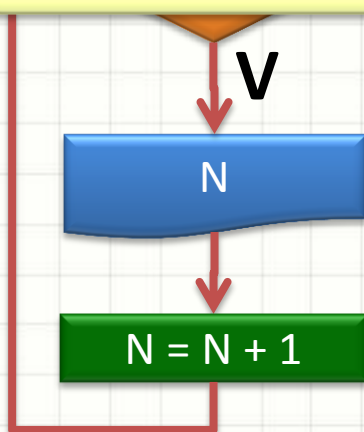
```
#include <iostream>
using namespace std;
main()
{
    int N;
    N = 3;
    while ( N <= 17 )
    {
        cout << N << endl;
        N = N + 1;
    }
}
```


Recordando o While

- Estrutura de repetição **while**

- O que faz?

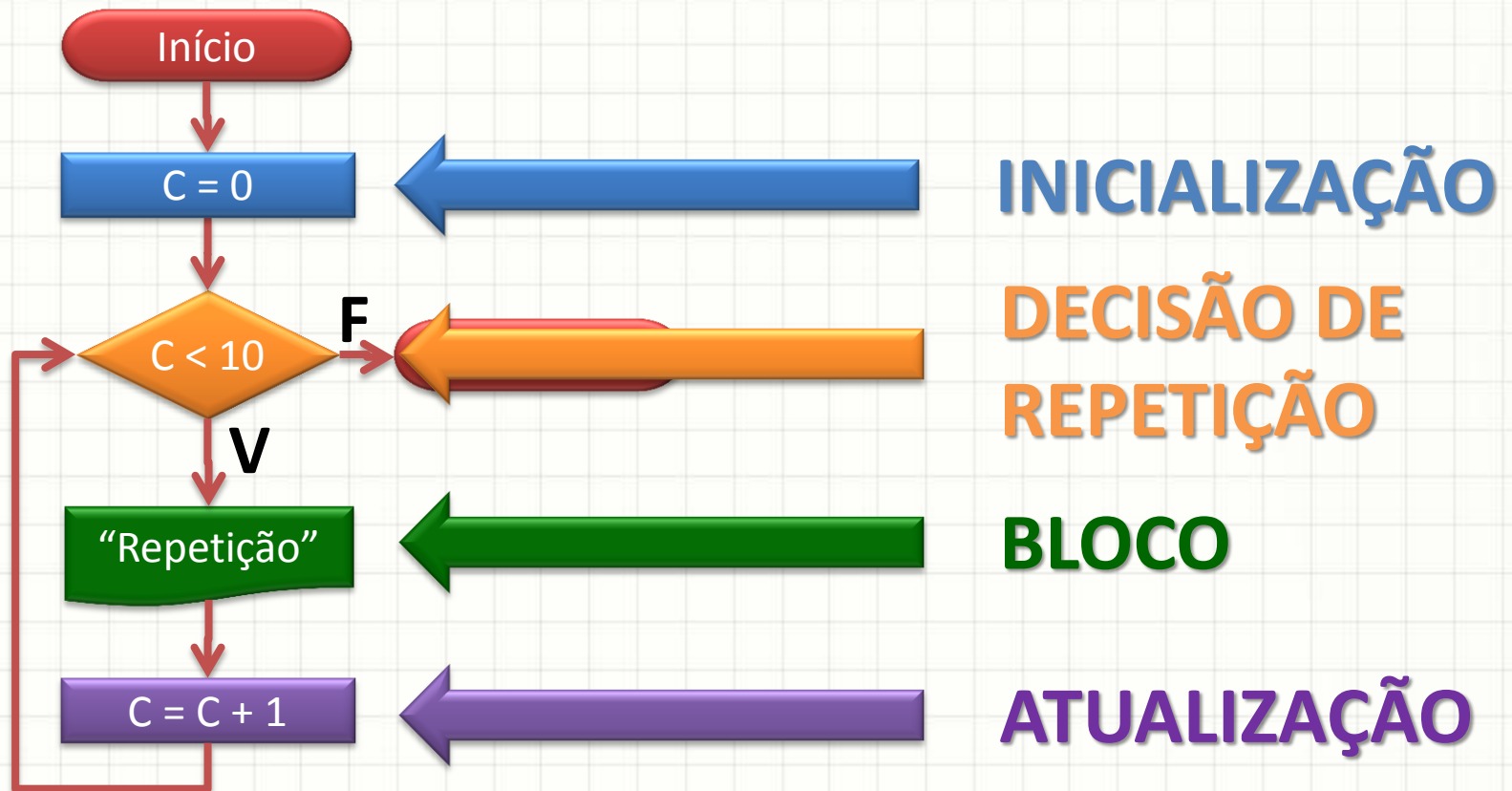
O que acontece se esquecermos essa linha?



```
#include <iostream>
using namespace std;
main()
{
    int N;
    N = 3;
    while ( N <= 17 )
    {
        cout << N << endl;
        N = N + 1;
    }
}
```

Recordando o While

- Observe:
 - O que faz?



Recordando o While

- No código...

```
#include <iostream>
using namespace std;
main()
```

```
{
```

```
int CONT;
```

```
CONT = 0;
```

```
while ( CONT < 10 )
```

```
{
```

```
cout << "Isso é uma Repetição" << endl;
```

```
CONT = CONT + 1;
```

```
}
```

```
}
```

INICIALIZAÇÃO

DECISÃO DE REPETIÇÃO

BLOCO

ATUALIZAÇÃO

Fácil esquecer um deles!

le

```
main()
```

```
{
```

```
int CONT;
```

```
CONT = 0;
```

```
while ( CONT < 10 )
```

```
{
```

```
cout << "Isso é uma Repetição" << endl;
```

```
CONT = CONT + 1;
```

```
}
```

```
}
```

INICIALIZAÇÃO

DECISÃO DE REPETIÇÃO

BLOCO

ATUALIZAÇÃO



A ESTRUTURA DE REPETIÇÃO FOR

O que é a estrutura **for**

- Todos os elementos em uma única linha
 - Só o bloco fica “isolado”

```
#include <iostream>
using namespace std;
main()
{
    int CONT;
    CONT = 0;
    while ( CONT < 10 )
    {
        cout << “Isso é uma Repetição” << endl;
        CONT = CONT + 1;
    }
}
```

O que é a estrutura **for**

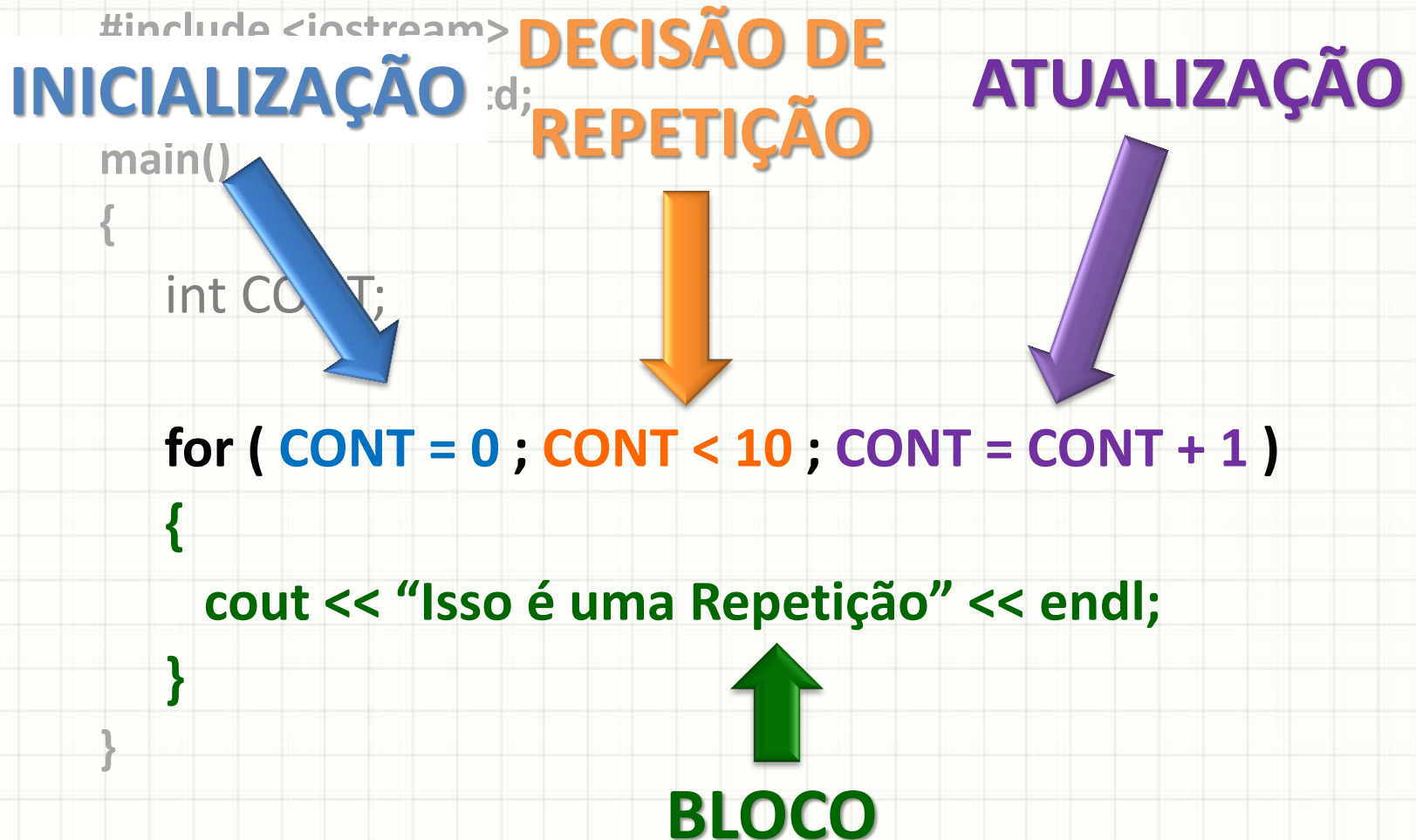
- Todos os elementos em uma única linha
 - Só o bloco fica “isolado”

```
#include <iostream>
using namespace std;
main()
{
    int CONT;

    for ( CONT = 0 ; CONT < 10 ; CONT = CONT + 1 )
    {
        cout << “Isso é uma Repetição” << endl;
    }
}
```

O que é a estrutura **for**

- Todos os elementos em uma única linha
 - Só o bloco fica “isolado”



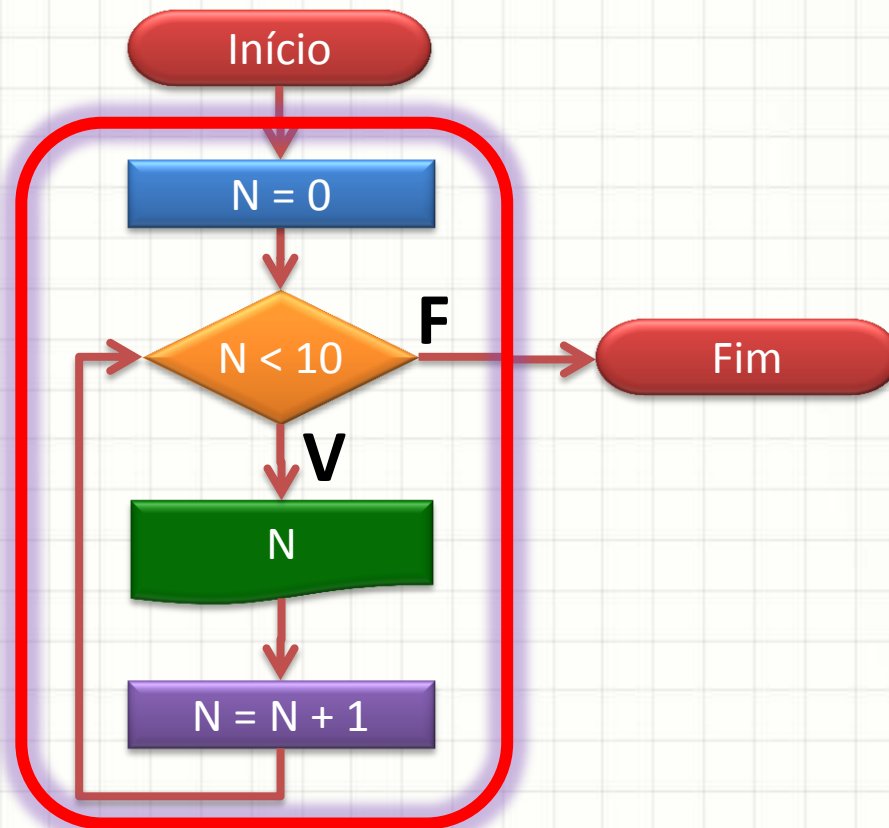
Forma Geral do for

```
for ( inicialização; condição de repetição; atualização)
```

```
{
```

Executa enquanto a proposição for verdadeira

```
}
```



EXERCÍCIO

A) Faça um programa que apresente seu nome 10.000 vezes.

EXERCÍCIO

A) Faça um programa que apresente seu nome 10.000 vezes.

```
#include <iostream>
using namespace std;
main()
{
    int C;

    for ( C = 0 ; C < 1000; C = C + 1 )
    {
        cout << "Daniel Caetano" << endl;
    }
}
```

EXERCÍCIO

B) Faça um programa que apresente os números de 52 a 75.

EXERCÍCIO

B) Faça um programa que apresente os números de 52 a 75.

```
#include <iostream>
using namespace std;
main()
{
    int C;

    for ( C = 52 ; C <= 75; C = C + 1 )
    {
        cout << C << endl;
    }
}
```

EXERCÍCIO

C) Modifique o programa anterior para que ele conte de 2 em 2.

EXERCÍCIO

C) Modifique o programa anterior para que ele conte de 2 em 2.

```
#include <iostream>
using namespace std;
main()
{
    int C;

    for ( C = 52 ; C <= 75; C = C + 2 )
    {
        cout << C << endl;
    }
}
```



MOMENTO LÚDICO:
O USO DE TABELAS

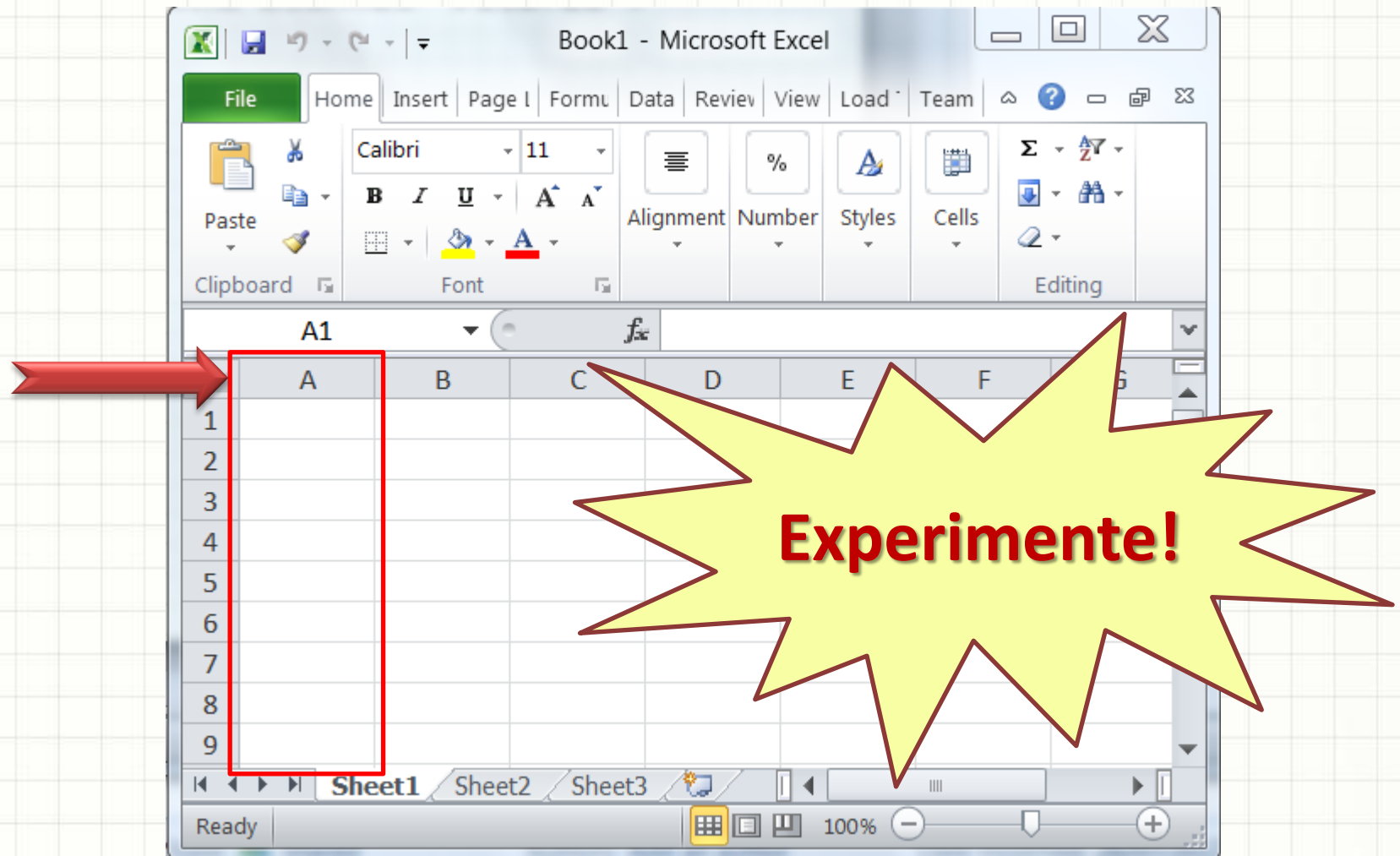
Momento Lúdico

- Em que programa usamos “tabelas”?



Momento Lúdico

- Em que programa usamos “tabelas”?





VETORES

Vetores

- Curso de Algoritmos:
 - Quantos valores se guardava em uma variável?
- Quantos valores posso guardar aqui?

int i;

- E nessa variável aqui?

float nota;

Vetores

- Mas e se quiséssemos guardar as notas de todos os alunos da turma (10 alunos)?

float nota1;

float nota2;

float nota3;

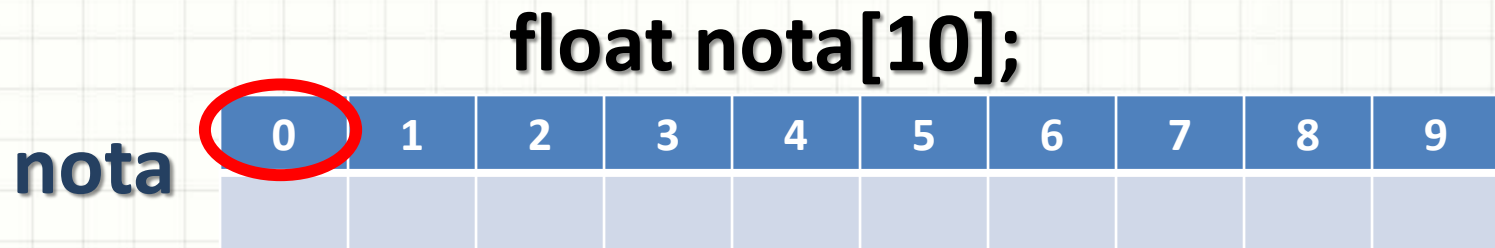
(...)

float nota10;

ARGH!

Vetores

- Guardando valores...



- Nota[0] = 10;

Vetores

- Guardando valores...

float nota[10];

nota	0	1	2	3	4	5	6	7	8	9
	10									

- Nota[0] = 10;

Vetores

- Guardando valores...

float nota[10];

nota	0	1	2	3	4	5	6	7	8	9
	10									

- Nota[0] = 10;
- Nota[5] = 7;

Vetores

- Guardando valores...

float nota[10];

nota	0	1	2	3	4	5	6	7	8	9
	10					7				

- Nota[0] = 10;
- Nota[5] = 7;

Exemplo

- Fazer um programa que leia três valores inteiros, duplique-os e depois os imprima.



EXERCÍCIOS

Vetores: Exercícios

1. Faça um programa que preencha um vetor de 15 posições chamado **tabela** com o valor 0 (zero) em todas as posições. O programa deve imprimir os valores do vetor.

Vetores: Exercícios

1. Faça um programa que preencha um vetor de 15 posições chamado **tabela** com o valor 0

```
#include <iostream>
using namespace std;

int main(void) {
    int tabela[15], i;

    for (i=0; i<=14; i++) tabela[i]=0;
    for (i=0; i<=14; i++) cout << tabela[i] << endl;
}
```

Vetores: Exercícios

2. Faça um programa que preencha um vetor de 15 posições chamado **tabela** com os valores de 1 a 15 nas posições de 0 a 14, respectivamente. O programa deve imprimir os valores do vetor.

Vetores: Exercícios

2. Faça um programa que preencha um vetor de 15 posições chamado **tabela** com os

```
#include <iostream>
using namespace std;

int main(void) {
    int tabela[15], i;

    for (i=0; i<=14; i++) tabela[i]=i+1;
    for (i=0; i<=14; i++) cout << tabela[i] << endl;
}
```


Vetores: Exercícios

3. Faça um programa que preencha um vetor de 10 posições chamado **dados** com valores **reais** digitados pelo usuário. Ao final da digitação, o programa deve imprimir o vetor.

Vetores: Exercícios

3. Faça um programa que preencha um vetor de 10 posições chamado **dados** com valores

```
#include <iostream>
using namespace std;

int main(void) {
    float dados[10];
    int i;

    for (i=0; i<=9; i++) {
        cout << "Digite o valor " << i+1 << ": ";
        cin >> dados[i];
    }
    for (i=0; i<=9; i++) cout << dados[i] << endl;
}
```



FUNÇÕES COM VETORES

Funções com Vetores

- Na aula passada, vimos as funções
- Tarefa “comum” → Funções
- **Já fizemos:** um programa que leia três valores inteiros, duplique-os e depois os imprima.
- Transformar o programa para funções
 - lerDados(vetor, quantidade)
 - dobrarDados(vetor, quantidade)
 - imprimirDados(vetor, quantidade)

Funções com Vetores

- Exercício: modificar o programa para ler valores em 3 vetores diferentes:
 - v, com 3 posições
 - w, com 4 posições
 - z, com 6 posições



EXERCÍCIO DE FIXAÇÃO

Funções: Exercícios

4. Modifique o programa para que a “limpeza” do vetor seja feita pela função indicada:

```
#include <iostream>
using namespace std;
```

```
int main(void) {
    int tabela[15], i;

    for (i=0; i<=14; i++) tabela[i]=0;
    for (i=0; i<=14; i++) cout << tabela[i] << endl;
}
```

```
void preenche(int vetor[], int n, int valor) {
    int i;
    for (i=0; i<n; i++) vetor[i]=valor;
}
```

Funções: Exercícios

```
#include <iostream>
using namespace std;

void preenche(int vetor[], int n, int valor);

int main(void) {
    int tabela[15], i;

    preenche(tabela, 15, 0);
    for (i=0; i<=14; i++) cout << tabela[i] << endl;
}

void preenche(int vetor[], int n, int valor) {
    int i;
    for (i=0; i<n; i++) vetor[i]=valor;
}
```


Funções: Exercícios

5. Agora modifique o programa 4 para que a impressão dos dados do vetor seja feita por uma função com o seguinte protótipo:

```
void imprime(int vetor[], int n)
```

Fu

5. A

imp

um

```
#include <iostream>
using namespace std;

void preenche(int vetor[], int n, int valor);
void imprime(int vetor[], int n);

int main(void) {
    int tabela[15], i;

    preenche(tabela, 15, 0);
    imprime(tabela, 15);
}

void imprime(int vetor[], int n) {
    int i;
    for (i=0; i<n; i++) cout << vetor[i] << endl;
}

void preenche(int vetor[], int n, int valor) {
    int i;
    for (i=0; i<n; i++) vetor[i]=valor;
}
```

Funções: Exercícios

6. Modifique o programa a seguir para que o preenchimento seja feito por uma função com o seguinte protótipo:

void preenche(int vetor[], int n, int vinicial)

```
#include <iostream>
using namespace std;

int main(void) {
    int tabela[15], i;

    for (i=0; i<=14; i++) tabela[i]=i+1;
    for (i=0; i<=14; i++) cout << tabela[i] << endl;
}
```

Exemplo de uso: **preenche(tabela, 15, 1);**

Funções: Exercícios

6. Modifique o programa a seguir para que o

```
#include <iostream>
using namespace std;

void preenche(int vetor[], int n, int valor);

int main(void) {
    int tabela[15], i;

    preenche(tabela, 15, 1);
    for (i=0; i<=14; i++) cout << tabela[i] << endl;
}

void preenche(int vetor[], int n, int vinicial) {
    int i;
    for (i=0; i<n; i++) vetor[i]=vinicial+i;
}
```

Funções: Exercícios

7. Dada uma sequência de n números, imprimi-la na ordem inversa à da leitura.

Entrada: N e depois a sequência.

Funções: Exercícios

8. Deseja-se publicar o número de acertos dos alunos em uma prova em forma de testes. A prova consta de 5 questões, cada uma com cinco alternativas identificadas por A, B, C, D e E.

Entrada:

a resposta de cada questão (gabarito)

o número de alunos da turma

todas as respostas de cada aluno

O programa responde o número de acertos de cada aluno assim que as respostas deste aluno forem digitadas.



PERGUNTAS?



CONCLUSÕES

Resumo

- Estruturas de Repetição
 - Repetição com “for”
 - Vetores e seu uso
 - Utilidade das funções com vetores
-
- Exercícios...!
 - E mais adiante... Listas sequenciais
 - Principais operações em listas sequenciais