

# Rodrigo Henrich

rodrigohenrich@faccat.br



# Lembram deste exercício?

Escreva um programa que leia 10 valores e conte quantos deles estão no intervalo entre 10 e 20. Depois escreva essa informação

Ele não pede para armazenar os números lidos, mas e se pedisse como fariam?

# Lembram deste exercício?

- Aqui eram apenas 10 números, o problema é quando são muitos números para armazenar ou não sabemos a quantidade de números
- Para isso podemos usar vetores

# Vetores em C

- Podemos ver um vetor como uma gaveta onde eu tenho várias variáveis do mesmo tipo dentro
- Para acessar determinada gaveta eu preciso saber a localização dela
- essa localização é dada por um índice numérico
- Também podemos chamá-los de **array**

A	D	J	K	B	T	O
---	---	---	---	---	---	---

# Vetores em C

- Então, vamos ver o índice do vetor

0	1	2	3	4	5	6
<b>A</b>	<b>D</b>	<b>J</b>	<b>K</b>	<b>B</b>	<b>T</b>	<b>O</b>

Como usar um vetor em C

# Criando um vetor em C

- Para declarar um vetor em C

tipo\_vetor nome\_vetor[tamanho]

Ex.

```
1  #include<stdio.h>
2  #include<locale.h>
3  int main(){
4      setlocale(LC_ALL, "");
5      int numero[10];
6  }
```

# Criando um vetor em C

- Criamos um vetor do tipo inteiro com 10 posições





# Criando um vetor em C

- Criamos um vetor do tipo inteiro com 10 posições
- Podemos popular ele com elementos

50	10	9	1	0	25	99	100	20	90
----	----	---	---	---	----	----	-----	----	----

# Criando um vetor em C

- Certo então, cada posição deste vetor tem um endereço

0	1	2	3	4	5	6	7	8	9
<b>50</b>	<b>10</b>	<b>9</b>	<b>1</b>	<b>0</b>	<b>25</b>	<b>99</b>	<b>100</b>	<b>20</b>	<b>90</b>

# Criando um vetor em C

- Como faço para acessar determinada posição do vetor?

0	1	2	3	4	5	6	7	8	9
<b>50</b>	<b>10</b>	<b>9</b>	<b>1</b>	<b>0</b>	<b>25</b>	<b>99</b>	<b>100</b>	<b>20</b>	<b>90</b>

# Criando um vetor em C

- Como faço para acessar determinada posição do vetor?
- O que temos na posição 4 do vetor?

0	1	2	3	4	5	6	7	8	9
<b>50</b>	<b>10</b>	<b>9</b>	<b>1</b>	<b>0</b>	<b>25</b>	<b>99</b>	<b>100</b>	<b>20</b>	<b>90</b>

# Criando um vetor em C

- Como faço para acessar determinada posição do vetor?
- O que temos na posição 4 do vetor?
- A posição 4 do vetor contém o número 0

0	1	2	3	4	5	6	7	8	9
50	10	9	1	0	25	99	100	20	90

- Essas posições do vetor são chamadas de **índice** do vetor, como já vimos

# Criando um vetor em C

- Em C para acessar determinada posição de um vetor, usamos

```
nome_vetor[indice] = valor;
```

ou

```
valor = nome_vetor[indice];
```

# Criando um vetor em C

- Exemplo

```
#include<stdio.h>
int main(){
    int numero[10];
    numero[0] = 50;
    numero[1] = 10;
    numero[2] = 9;
    numero[3] = 1;
    numero[4] = 0;
    numero[5] = 25;
    numero[6] = 99;
    numero[7] = 100;
    numero[8] = 20;
    numero[9] = 90;
    int valor = numero[4];
    printf("%d", valor);
}
```

# Criando um vetor em C

- Aqui eram apenas 10 números, o problema é quando são muitos números para armazenar ou não sabemos a quantidade de números
- Agora temos uma maneira melhor de armazenar esses números, qual é ela?



# Criando um vetor em C

- Inicializar um vetor...
- Para isso usamos o formato de conjunto numérico

```
#include<stdio.h>
int main(){
    int numero[10] = {50,10,9,1,0,25,99,100,20,90};
    int valor = numero[4];
    printf("%d", valor);
}
```

# Criando um vetor em C

- Pensando nas estruturas de repetição, podemos usar uma delas para percorrer um vetor.
- Alguns exemplos....

# Criando um vetor em C

- Percorrendo o vetor usando for

```
1  #include<stdio.h>
2  #include<locale.h>
3  int main(){
4      setlocale(LC_ALL, "");
5      int vetor[10];
6      for(int i=0;i<10;i++){
7          printf("\nDigite um número para posição %d do vetor: ",i);
8          scanf("%d",&vetor[i]);
9      }
10     printf("\nDados do vetor...\n");
11     for(int i=0;i<10;i++)
12         printf("%d ", vetor[i]);
13 }
```

# Criando um vetor em C

- Percorrendo o vetor usando while

```
1  #include<stdio.h>
2  #include<locale.h>
3  int main(){
4      setlocale(LC_ALL, "");
5      int vetor[10];
6      int i=0;
7      while(i<10){
8          printf("\nDigite um número para posição %d do vetor: ",i);
9          scanf("%d",&vetor[i]);
10         i++;
11     }
12     printf("\nDados do vetor...\n");
13     i=0;
14     while(i<10)
15         printf("%d ", vetor[i++]);
16 }
```

# E se eu não usar todas as posições do vetor

- Em alguns momentos pode ocorrer a leitura de lixo do buffer do teclado
- Com isso algumas leituras de dados podem ser ignoradas
- A função **setbuf(stdin, NULL)**
- Escreve null dentro do buffer do teclado
- Desta forma evitamos que nosso programa tenha um comportamento errado

# E se eu não usar todas as posições do vetor

- Usando um contador para gerenciar a ocupação do vetor
- Podemos declarar um contador e a cada novo elemento que adicionar ao vetor incrementamos ele
- Assim não é necessário preencher o vetor completamente
- Vamos ver um exemplo usando do...while

# E se eu não usar todas as posições do vetor

```
1  #include<stdio.h>
2  #include<locale.h>
3  int main(){
4      setlocale(LC_ALL, "");
5      int vetor[10];
6      int cont = 0;
7      char resposta = 's';
8      do{
9          printf("\nDigite um número para posição %d do vetor: ", cont);
10         scanf("%d",&vetor[cont]);
11         cont++;
12         printf("Continuar? s/n");
13         setbuf(stdin, NULL);
14         resposta = getchar();
15     }while(resposta != 'n' && cont < 10);
16 }
```

# E se eu não usar todas as posições do vetor

- Certo e para mostrar os dados percorro todo o vetor?



# E se eu não usar todas as posições do vetor

- Não
- Temos que percorrer até cont;

```
for(int i=0;i<cont;i++)  
    printf("%d ",vetor[i]);
```

# Gerar números aleatórios em C

- Para gerar um número aleatório em C usamos a função `rand()`
- Para usar essa função é preciso incluir a biblioteca `stdlib.h`
- A sintaxe dela é

`rand() % limite_máximo`

# Gerar números aleatórios em C

```
1  #include<stdio.h>
2  #include<locale.h>
3  | #include<stdlib.h>
4  | int main(){
5  |     setlocale(LC_ALL, "");
6  |     printf("Número aleatório %d", rand()%100);
7  | }
```

# Gerar números aleatórios em C

- Se rodar esse programa mais de uma vez verá que ele gera sempre o mesmo número aleatório
- O que torna ele não tão aleatório assim...
- Para resolver isso temos que carregar a função `rand()` com algo aleatório
- Vamos usar o time da máquina para isso usando a função `srand()` e a função `time()`;
- Para função `time()` precisamos adicionar a biblioteca `time.h`

# Gerar números aleatórios em C

```
1  #include<stdio.h>
2  #include<locale.h>
3  #include<stdlib.h>
4  #include<time.h>
5  int main(){
6      setlocale(LC_ALL,"");
7      srand(time(NULL));
8      printf("Número aleatório %d",rand()%100);
9  }
```