



PARADIGMAS DE LINGUAGENS DE PROGRAMAÇÃO EM PYTHON

TIPOS DE DADOS AVANÇADOS I

Prof. Dr. Daniel Caetano

2022 - 1

Compreendendo o problema

- **Situação:** sistema da universidade: matrícula e média
 - Não sabemos o total de alunos, mas sabemos o máximo...

Aluno

- Matrícula (int)
- Média (float)

- Em C/C++... Como declarar?



Mentimeter

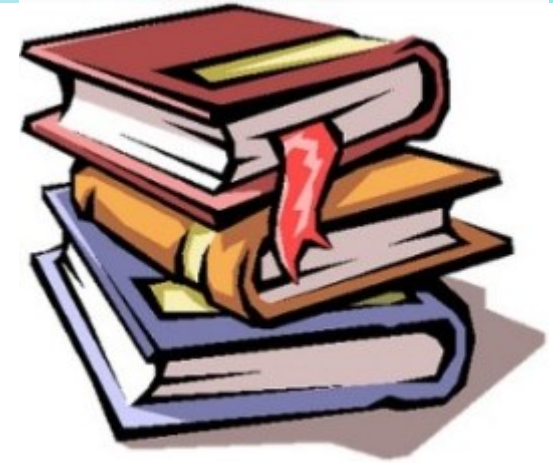
<https://www.menti.com/>

Objetivos

- Compreender o processo de criação de funções
- Compreender o conceito de parâmetro
- Compreender o conceito de retorno
- Compreender a lógica das listas estáticas



Bibliografia da Aula



Material	Acesso ao Material
Apresentação	https://www.caetano.eng.br/aulas/2022a/ara0066.php (Paradigmas de Programação – Aula 08)
Livro Texto	Capítulo 6, páginas 273 a 287
Aprenda Mais!	<ul style="list-style-type: none">• Vídeo: Listas Encadeadas (ative legenda traduzida!) https://youtu.be/njTh_OwMljA

FUNÇÕES



Mentimeter

<https://www.menti.com/>

Subprogramas em Python

- Já vimos antes...

```
def assina():  
    print("Atenciosamente,")  
    print("Prof. Daniel Caetano")  
    print("prof@caetano.eng.br")  
  
assina()
```

- Será que em C/C++ é tão simples?
 - Quase a mesma coisa...

Subprogramas em C/C++

Exemplo 01

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

void assina() {
    cout << "Atenciosamente," << endl;
    cout << "Prof. Daniel Caetano" << endl;
    cout << "prof@caetano.eng.br" << endl;
}

int main() {
    assina();
}
```

Por que chamamos alguns subprogramas de funções?

Momento Lúdico

- Como fazer um misto quente?
- Como fazer um sanduiche com um “recheio” genérico?



Funções: subprogramas + parâmetros

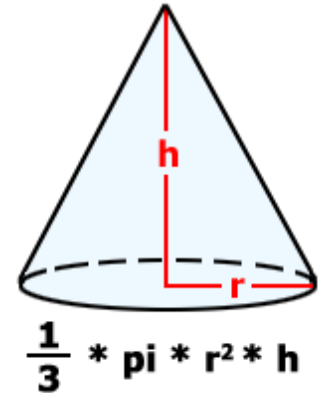
- Calcular o volume de um cone

Exemplo 02

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

```
void volumeCone(float r, float h) {
    float res = (1.0/3.0)*3.14*r*r*h;
    cout << res;
}
```

```
main() {
    volumeCone(10,2);
}
```



Funções: subprogramas + parâmetros

- Calcular o volume de um cone

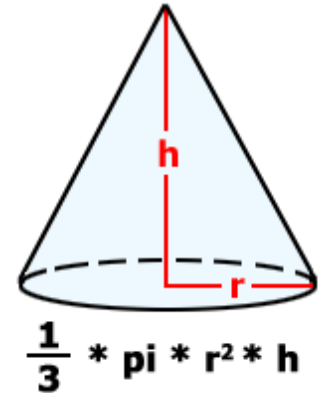
Exemplo 02

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

```
void volumeCone(float r, float h) {
    float res = (1.0/3.0)*3.14*r*r*h;
    cout << res;
}
```

Mude esses valores e teste!

```
main() {
    volumeCone(10,2);
}
```



Funções: subs + parâms + retorno

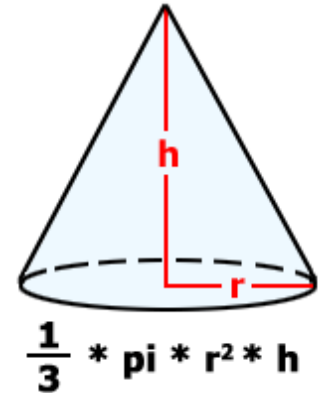
- Calcular o volume de um cone

Exemplo 02b

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

```
float volumeCone(float r, float h) {
    float res = (1.0/3.0)*3.14*r*r*h;
    return res;
}
```

```
main() {
    cout << volumeCone(10,2);
}
```



Funções: subs + parâms + retorno

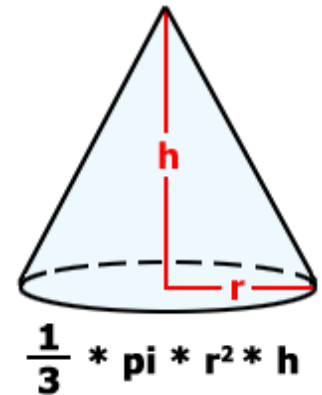
- Calcular o volume de um cone

Exemplo 02b

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

```
float volumeCone(float r, float h) {
    float res = (1.0/3.0)*3.14*r*r*h;
    return res;
}
```

```
main() {
    cout << volumeCone(10,2);
}
```



Flexibilizar
Organizar
Reaproveitar

Funções: Modificando Parâmetros

- Função que atualiza salário (+10%)

Exemplo 03

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

void atualizaSalario(float sal) {
    sal = sal * 1.10;
}

int main() {
    float s=100.00 ;
    atualizaSalario(s);
    cout << s;
}
```

Funções: Modificando Parâmetros

- Função que atualiza salário (+10%)

Exemplo 03

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

void atualizaSalario(float sal) {
    sal = sal * 1.10;
}

int main() {
    float s=100.00 ;
    atualizaSalario(s);
    cout << s;
}
```

Não funciona!

Funções: Modificando Parâmetros

- Função que atualiza salário (+10%)

Exemplo 03

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

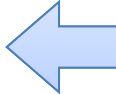
Passagem por
referência



```
void atualizaSalario(float &sal) {
    sal = sal * 1.10;
}
```

```
int main() {
    float s=100.00 ;
    atualizaSalario(s);
    cout << s;
}
```

E se colocar um
número aqui?





FUNÇÕES: ATIVIDADES

Atividade 01 – Solução no Tutorial

Crie (e use) uma função que imprima o texto

Eu estou funcionando!

https://www.tutorialspoint.com/compile_cpp_online.php

Atividade 02 – Solução no Tutorial

Crie (e use) uma função que some dois valores inteiros e retorne o resultado.

Atividade 03 – Solução no Tutorial

Crie uma função em C/C++ que calcule a área de um retângulo e tenha a seguinte estrutura:

```
double calculaArea(double base, double altura)
```

Depois construa a **main** que usa essa função

LISTAS LINEARES ESTÁTICAS



<https://www.menti.com/>

Listas Lineares Estáticas

- Como representar a bibliografia do curso?
- Como representar os contatos telefônicos?
- Como representar o conjunto de notas dos alunos?

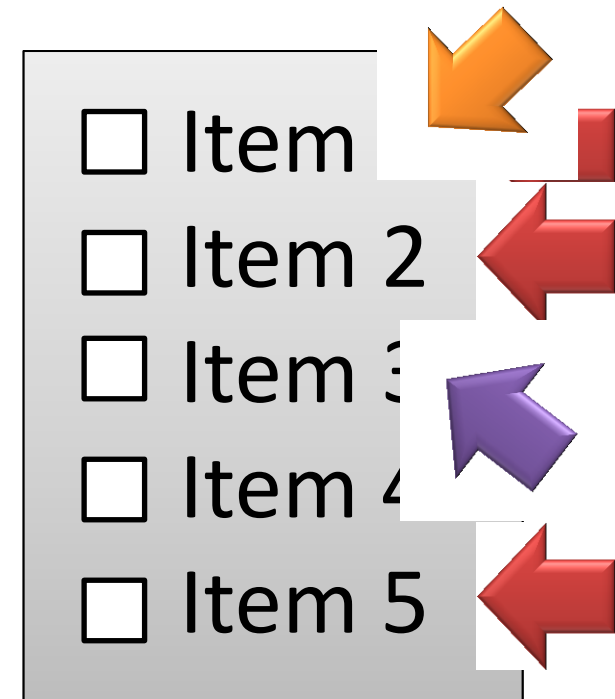
Lista

**Só que esse não
é tipo primitivo
do C/C++!**



Entendendo Detalhes da Lista

- Uma lista é um conjunto de elementos
 - Usualmente de um mesmo tipo
 - Possui uma ordem
 - Primeiro elemento
 - Último elemento
 - Elementos Intermediários
 - Antecessor
 - Sucessor



Listas: Detalhes de Implementação

- Se sabemos o tamanho máximo da lista...
 - Podemos alocar todo o espaço...
 - Espaço “contíguo” na memória
 - Podemos usar um vetor!
- Exemplo: armazenar **até** 10 notas de alunos

float notas[10];

Listas: Detalhes de Implementação

- Por que podemos usar vetor?
 - Tamanho máximo da lista: **10**
 - Dados todos do mesmo tipo: **float**
- A lista vai estar sempre cheia?
 - Se houver só 7 notas, quantas imprimir?
 - Mas como vamos saber que são 7?
 - Variável de controle de *quantidade*

```
float notas[10];  
int quantidade;
```


Listas: Detalhes de Implementação

- Note: estrutura composta por **duas** variáveis

```
float notas[10];  
int quantidade;
```

Listas: Detalhes de Implementação

- Vamos começar um programa e declarar nossa lista dentro do main

```
float notas[40];
```

```
int quantidade;
```

nota:

0	1	2	3	4	5	6	7	8	9
?	?	?	?	?	?	?	?	?	?

quantidade: ???

Listas: Detalhes de Implementação

- Que operações podemos implementar?
 - *Inicializar*
 - *Inserir*
 - *Listar*
 - *Buscar*
 - *Remover*
 - *Substituir*
 - *Ordenar*
 - ...

Listas: Detalhes de Implementação

- Que operações podemos implementar?
 - *Inicializar*
 - *Inserir*
 - *Listar*
 - *Buscar*
 - *Remover*
 - *Substituir*
 - *Ordenar*
 - ...

Estruturas de
Dados!

Listas: Lógica de Criação

- Inicializar?
 - Definir o “status” inicial
 - Prepará-la para o uso
 - O que caracteriza uma lista que não recebeu dados?
- Vamos implementar a inicialização?

nota:

0	1	2	3	4	5	6	7	8	9
?	?	?	?	?	?	?	?	?	?

quantidade: 0

Exemplo 04

Listas: Lógica de Inserção

- Inserir
 - Vamos considerar uma lista sem ordenação
 - Vamos considerar que pode haver repetição
 - Acrescentar valor no fim do vetor
 - Verificar se está cheia!

nota:



0
?

quantidade: 0

3	4	5	6	7	8	9
?	?	?	?	?	?	?

- Depois de inserir dado, incrementar *quantidade*

Listas: Lógica de Inserção

- Inserir
 - Vamos implementar a função inserir?
 - Parâmetros
 - Vetor
 - Dado a inserir
 - Posição (= quantidade) **por referência!**
 - Tamanho máximo da lista
- Vamos ler 1 valor na **main** e inseri-lo na lista

Exemplo 05a

Exemplo de Uso

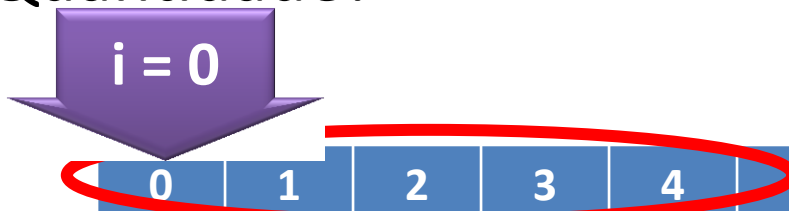
- Implementar, na main, a leitura de vários valores, até o usuário digitar o valor 0
 - Cada valor lido deve ser inserido na lista
 - Se o valor for zero, ele não deve ser inserido

Exemplo 05b

Listas: Lógica de Varredura

- Listar?
 - Imprimir um a um os valores...
 - Até o fim do vetor?
 - Quantidade!

nota:



0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?

quantidade: 5

Listas: Lógica de Varredura

- Listar?
 - Imprimir um a um os valores...
 - Até o fim do vetor?
 - Quantidade!

nota:

0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?

quantidade: 5

Listas: Lógica de Varredura

- Listar?
 - Imprimir um a um os valores...
 - Até o fim do vetor?
 - Quantidade!

nota:

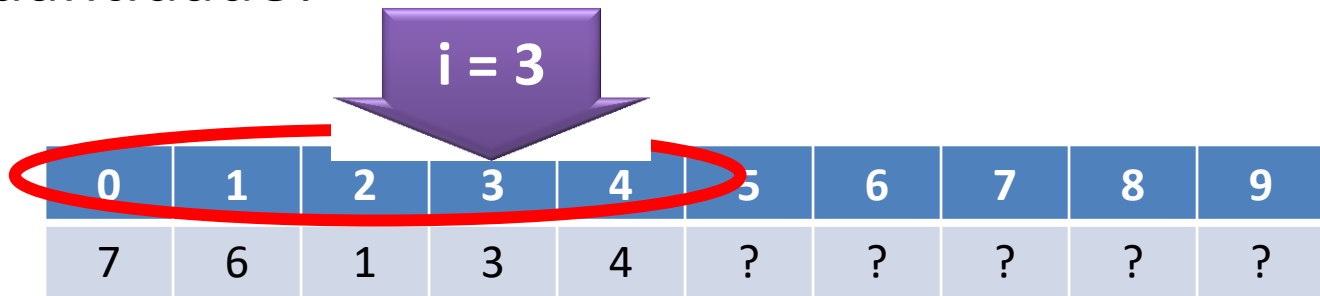
0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?

quantidade: 5

Listas: Lógica de Varredura

- Listar?
 - Imprimir um a um os valores...
 - Até o fim do vetor?
 - Quantidade!

nota:



0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?

quantidade: 5

Listas: Lógica de Varredura

- Listar?
 - Imprimir um a um os valores...
 - Até o fim do vetor?
 - Quantidade!

nota:

0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?

quantidade: 5

- $i = 0$ enquanto $i < \text{quantidade}$

Exemplo de Aplicação


- Listar
 - Vamos implementar a função listar?
 - Parâmetros
 - Vetor
 - Quantidade
- Vamos imprimir o vetor na **main**?

Exemplo 06

Listas: Lógica de Busca

- Buscar?
 - Procurar por um valor
 - Até o fim do vetor?
 - Quantidade!

nota:



0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?


quantidade: 5

valor: 1

Listas: Lógica de Busca

- Buscar?
 - Procurar por um valor
 - Até o fim do vetor?
 - Quantidade!

nota:

									
0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?


quantidade: 5

valor: 1

Listas: Lógica de Busca

- Buscar?
 - Procurar por um valor
 - Até o fim do vetor?
 - Quantidade!

nota:



0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?

quantidade: 5

valor: 1

YES!!

Listas: Lógica de Busca

- Buscar
 - Mas e se a busca não encontrar o número?
 - Responder **-1**
 - **Por quê?**
 - Parâmetros
 - Vetor
 - Quantidade
 - Valor

Exemplo de Aplicação


- Vamos implementar a função **buscar**!
- Modificar a **main** para permitir uma busca
 - Se encontrar, deve imprimir a posição
 - Se não encontrar, deve imprimir que não achou

Exemplo 07

Listas: Lógica de Remoção

- Remove?
 - Remover um dado valor
 - Procurar por ele...
 - Copiar o último elemento sobre ele

nota:



0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?


quantidade: 5

valor: 1

Listas: Lógica de Remoção

- Remove
 - Remove um dado valor
 - Procurar por ele...
 - Copiar o último elemento sobre ele

nota:



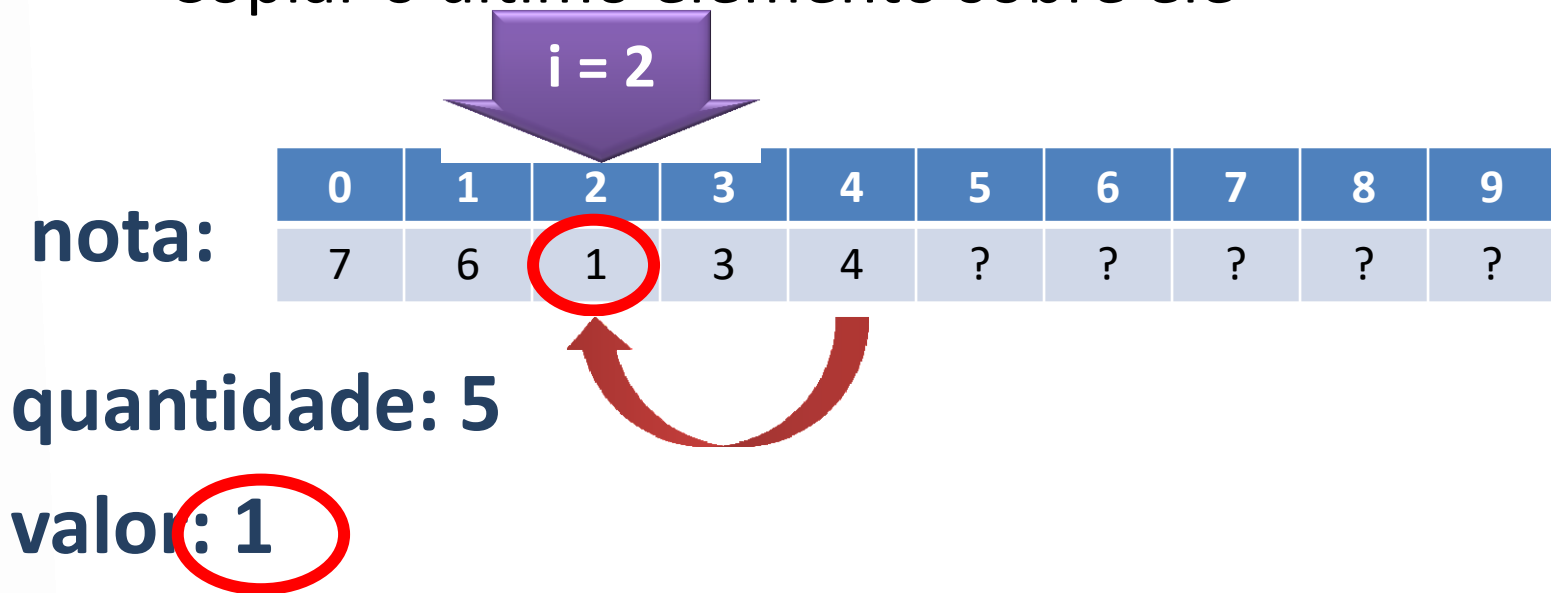
0	1	2	3	4	5	6	7	8	9
7	6	1	3	4	?	?	?	?	?

quantidade: 5

valor: 1

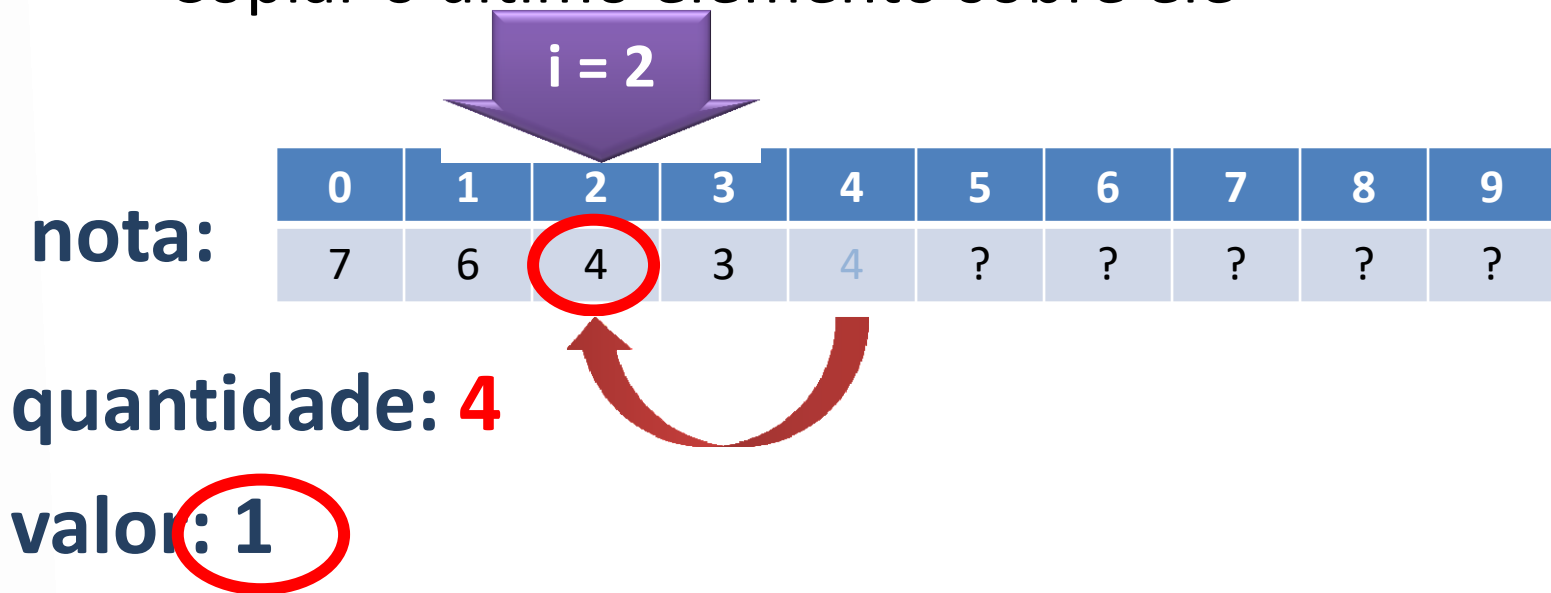
Listas: Lógica de Remoção

- Remove
 - Remove um dado valor
 - Procurar por ele...
 - Copiar o último elemento sobre ele



Listas: Lógica de Remoção

- Remove
 - Remove um dado valor
 - Procurar por ele...
 - Copiar o último elemento sobre ele



Listas: Lógica de Remoção

- Remove
 - Deve imprimir erro se
 - Lista vazia
 - Elemento não encontrado
 - Parâmetros
 - Vetor
 - Quantidade **por referência!**
 - Valor

Exemplo de Aplicação

- Vamos implementar a função **remove**!
- Modifique a **main** para permitir que um elemento seja removido
 - Imprimir a lista após remoção

Exemplo 08



LISTAS ESTÁTICAS: ATIVIDADES

Atividade 04 – Solução no Tutorial

- 10 minutos
- Faça um programa para receber uma lista de até 50 **inteiros** e depois imprimir a lista dos números digitados (*use como base o código fornecido pelo professor!*)
 - A inserção deve parar se um número negativo for digitado
 - O número negativo não deve ser inserido na lista
 - Imprima a lista ao final.

Atividade 05 – Solução no Tutorial

- 10 minutos
- Altere o programa anterior para que ele receba uma lista de até 50 inteiros **distintos** e depois imprimir a lista dos números digitados
- Para esse programa, crie uma função

void inserirSemRepetir(int v[], int valor, int &pos, int max)

Atividade 06 – Solução no Tutorial

- 5 minutos
- Modifique o programa anterior para que ele guarde até 50 números de telefone (long) distintos.
 - A inserção deve parar se um número com menos de 8 dígitos (menor que 10000000) for digitado
 - Números com menos de 8 dígitos não devem ser guardados



VOLTANDO AO PROBLEMA ORIGINAL: EXEMPLO

Compreendendo o problema

- **Situação:** sistema da universidade: matrícula e média
 - Não sabemos o total de alunos, mas sabemos o máximo...

Aluno

- Matrícula (int)
- Média (float)

- Em C/C++... Como declarar?

Exemplo 09

- Modifique o programa para que ele trabalhe com uma lista de Alunos, descritos pela seguinte estrutura:

Aluno

- Matrícula (int)
- Média (float)



ENCERRAMENTO

Resumo e Próximos Passos

- Funções
 - Parâmetros e Retorno
 - Listas Estáticas
 - Operações!
 - **Pós Aula:** Saiba Mais, A Seguir... e Desafio!
 - No mural: <https://padlet.com/djcaetano/paradigmas>
-
- Todas as listas são estáticas?
 - Como fazer a lista crescer quando necessário?



PERGUNTAS?