



INFORMÁTICA PARA ENGENHARIA

LISTAS, VETORES, FILAS E PILHAS

Prof. Dr. Daniel Caetano

2019 - 2

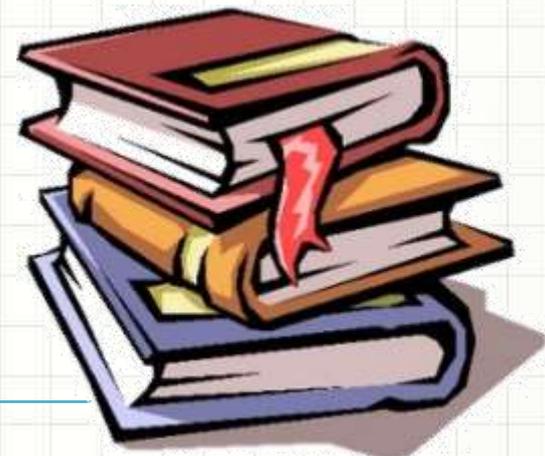
Objetivos

- Compreender a ideia das listas
- Compreender o uso das listas unidimensionais: os vetores
- Compreender as filas e pilhas

- **Exercícios Aula 13!**

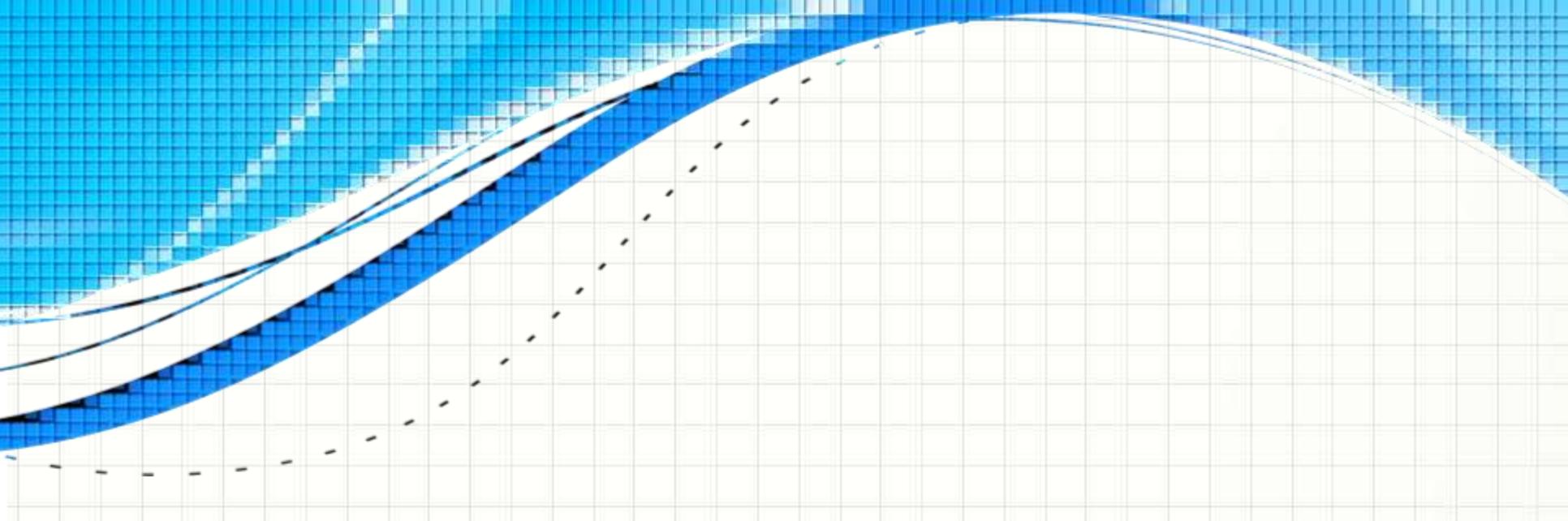


Material de Estudo



Material	Acesso ao Material
Notas de Aula e Apresentação	http://www.caetano.eng.br/ (Informática para Engenharia – Aula 13)
Material Didático	Lógica de Programação, págs 151 a 172.
Biblioteca Virtual	“Lógica de Programação – Fundamentos da Programação de Computadores”, págs 93 a 144.
Material Adicional	https://www.ime.usp.br/~leo/mac2166/2017-1/introducao_vetores_python.html Filas e Pilhas em Python: https://tinyurl.com/yxmyof47 Estruturas de Dados: https://tinyurl.com/y3smldc6

LEMBRETE: CONSULTAR O “DEPOIS” DA AULA 13 NO SAVA!



COMPREENDENDO A SITUAÇÃO PRÁTICA

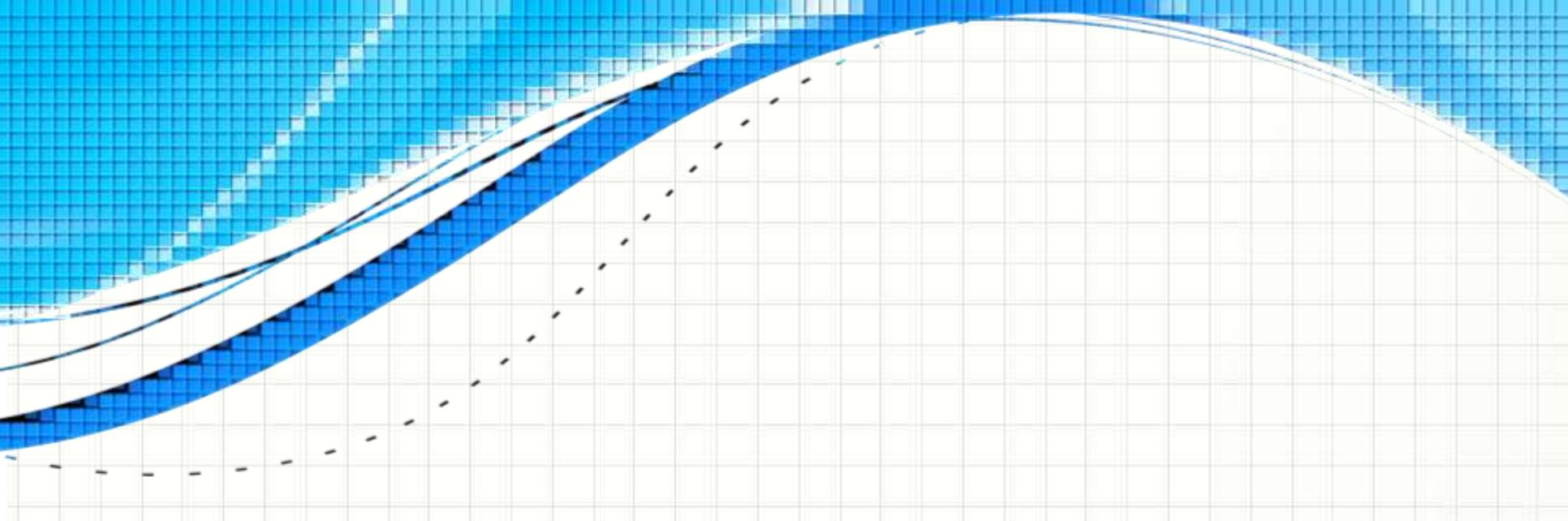
Quando usar uma lista?

- Problema: ler 3 nomes e listar aqueles cujo número de letras é maior que a média

```
# Lista nomes maiores que a média
NOME1 = input("Digite 1º nome: ")
NOME2 = input("Digite 2º nome: ")
NOME3 = input("Digite 3º nome: ")
MED = (len(NOME1) + len(NOME2) + len(NOME2))/3
if len(NOME1) > MED :
    print (NOME1)
if len(NOME2) > MED :
    print (NOME2)
if len(NOME3) > MED :
    print (NOME3)
```

- Qual o problema dessa solução?
 - E se fossem 50 nomes?

Solução: Listas



O QUE É UMA LISTA PARA O PYTHON?

O que é uma lista?

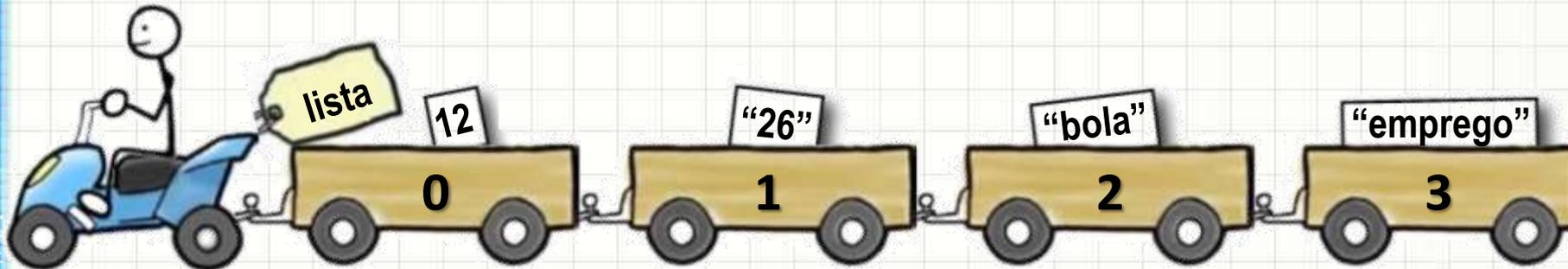
- É um conjunto de dados encadeados
 - Uma lista é como um trem de dados:

Cada vagão guarda uma informação!



O que é uma lista?

- **lista** é uma única variável, com vários dados
 - Os vagões são numerados



```
In[1]: print( lista[1] )
```

```
26
```

```
In[2]: print( lista[3] )
```

```
emprego
```

As listas são também chamadas de **vetores**

Criando uma lista no console

- Podemos criar uma lista vazia:

Console

```
In [1]: lista = []
```

**Observe o
explorador de
variáveis!**

`lista`

`list`

`0`

`[]`

Criando uma lista no console

- Podemos criar uma lista já preenchida:

Console

```
In [2]: lista = [12, "26", "bola", "emprego"]
```

**Observe o
explorador de
variáveis!**

lista	list	4	[12, '26', 'bola', 'emprego']
-------	------	---	-------------------------------

Criando uma lista no console

- Podemos mostrar o conteúdo da lista:

Console

```
In [3]: print( lista )
```

**O que
aconteceu?**

Usando uma lista no console

- Podemos obter o tamanho da lista

Console

```
In [4]: N = len( lista )
```

```
In [5]: print( N )
```

**O que
aconteceu?**

Usando uma lista no console

- Podemos acrescentar um item na lista

Console

```
In [6]: lista.append( 33 )
```

**Observe o
explorador de
variáveis!**

<code>lista</code>	<code>list</code>	<code>5</code>	<code>[12, '26', 'bola', 'emprego', 33]</code>
--------------------	-------------------	----------------	------------------------------------------------

Usando uma lista no console

- Podemos mudar um valor da lista

Console

```
In [7]: print ( lista[3] )
```

```
emprego
```

```
In [8]: lista[3] = "praia"
```

```
In [9]: print ( lista[3] )
```

```
praia
```

Usando uma lista no console

- Podemos remover um elemento da lista

Console

```
In [10]: print ( lista )
```

```
[12, '26', 'bola', 'praia', 33]
```

```
In [11]: lista.remove( "26" )
```

```
In [12]: print ( lista )
```

```
[12, 'bola', 'praia', 33]
```

Usando uma lista no console

- Podemos remover um elemento da lista

Console

```
In [13]: print ( lista )  
[12, 'bola', 'praia', 33]  
In [14]: lista.pop( 1 )  
In [15]: print ( lista )  
[12, 'praia', 33]
```

Usando uma lista no console

- Podemos juntar listas

Console

```
In [16]: print ( lista )
```

```
[12, 'praia', 33]
```

```
In [17]: lista2 = [ "Mais", "Elementos" ]
```

```
In [18]: lista = lista + lista2
```

```
In [19]: print ( lista )
```

```
[12, 'praia', 33, 'Mais', 'Elementos' ]
```

Exercício

- Complete o programa abaixo para que ele faça a média das notas da lista

```
aula13ex01.py
```

```
# Calcula a média das notas
```

```
NOTAS = [ 8.0, 5.5, 7.5, 6.0, 3.2 ]
```

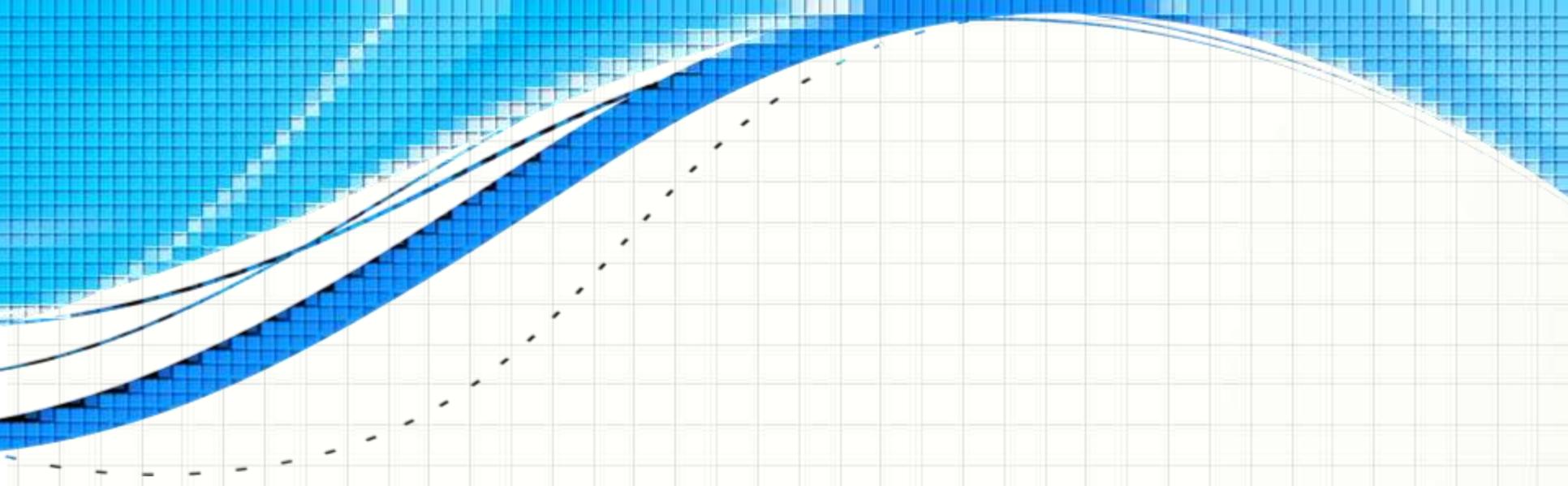
Exercício

- Complete o programa abaixo para que ele faça a média das notas da lista

```
aula13ex01.py
```

```
# Calcula a média das notas

NOTAS = [ 8.0, 5.5, 7.5, 6.0, 3.2 ]
SOMA = 0
for NOTA in NOTAS:
    SOMA = SOMA + NOTA
MEDIA = SOMA / len(NOTAS)
print( "A média é {:.1f}.".format(MEDIA))
```



**VOLTANDO AO
PROBLEMA ORIGINAL**

Problema Original

- Problema: ler 3 nomes e listar aqueles cujo número de letras é maior que a média

```
# Lista nomes maiores que a média
NOME1 = input("Digite 1º nome: ")
NOME2 = input("Digite 2º nome: ")
NOME3 = input("Digite 3º nome: ")
MEDIA = (len(NOME1) + len(NOME2) + len(NOME2))/3
if len(NOME1) > MEDIA :
    print (NOME1)
if len(NOME2) > MEDIA :
    print (NOME2)
if len(NOME3) > MEDIA :
    print (NOME3)
```

- Como resolver usando listas?

Problema Original – Com Lista

```
# Lista nomes maiores que a média
NOMES = []
# Lê os nomes
for N in range(3) :
    NOME = input("Digite um nome: ")
    NOMES.append( NOME )
# Computa o tamanho médio dos nomes
TOTAL = 0
for NOME in NOMES :
    TOTAL = TOTAL + len(NOME)
MEDIA = TOTAL/len(NOMES)
# Imprime aqueles cujo tamanho é maior que a média
for NOME in NOMES :
    if len(NOME) > MEDIA :
        print (NOME)
```

Melhorando?

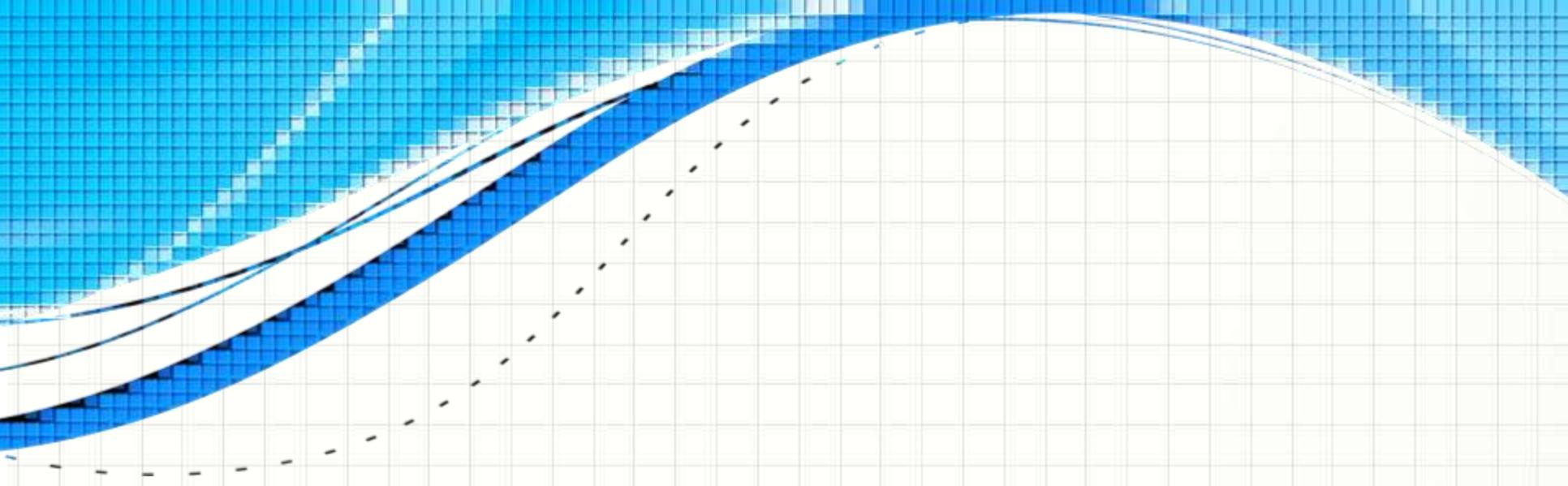
Problema Original – Com Lista

```
# Lista nomes maiores que a média
NOMES = []
# Lê os nomes
for N in range(3) :
    NOME = input("Digite o " + str(N+1) + "º nome: ")
    NOMES.append( NOME )
# Computa o tamanho médio dos nomes
TOTAL = 0
for NOME in NOMES :
    TOTAL = TOTAL + len(NOME)
MEDIA = TOTAL/len(NOMES)
# Imprime aqueles cujo tamanho é maior que a média
for NOME in NOMES :
    if len(NOME) > MEDIA :
        print (NOME)
```

E para 10 nomes?

Problema Original – Com Lista

```
# Lista nomes maiores que a média
NOMES = []
# Lê os nomes
for N in range(10) :
    NOME = input("Digite o " + str(N+1) + "º nome: ")
    NOMES.append( NOME )
# Computa o tamanho médio dos nomes
TOTAL = 0
for NOME in NOMES :
    TOTAL = TOTAL + len(NOME)
MEDIA = TOTAL/len(NOMES)
# Imprime aqueles cujo tamanho é maior que a média
for NOME in NOMES :
    if len(NOME) > MEDIA :
        print (NOME)
```

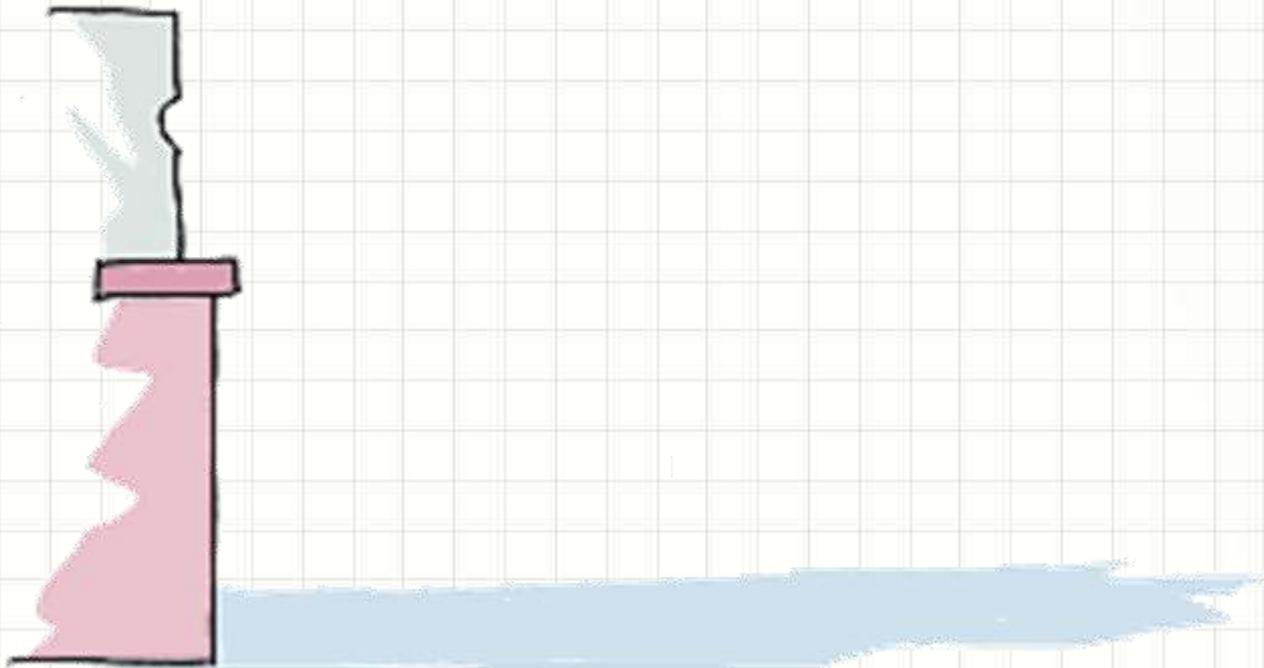


FILAS:

LISTAS COM REGRAS CLARAS PARA ENTRADA E SAÍDA

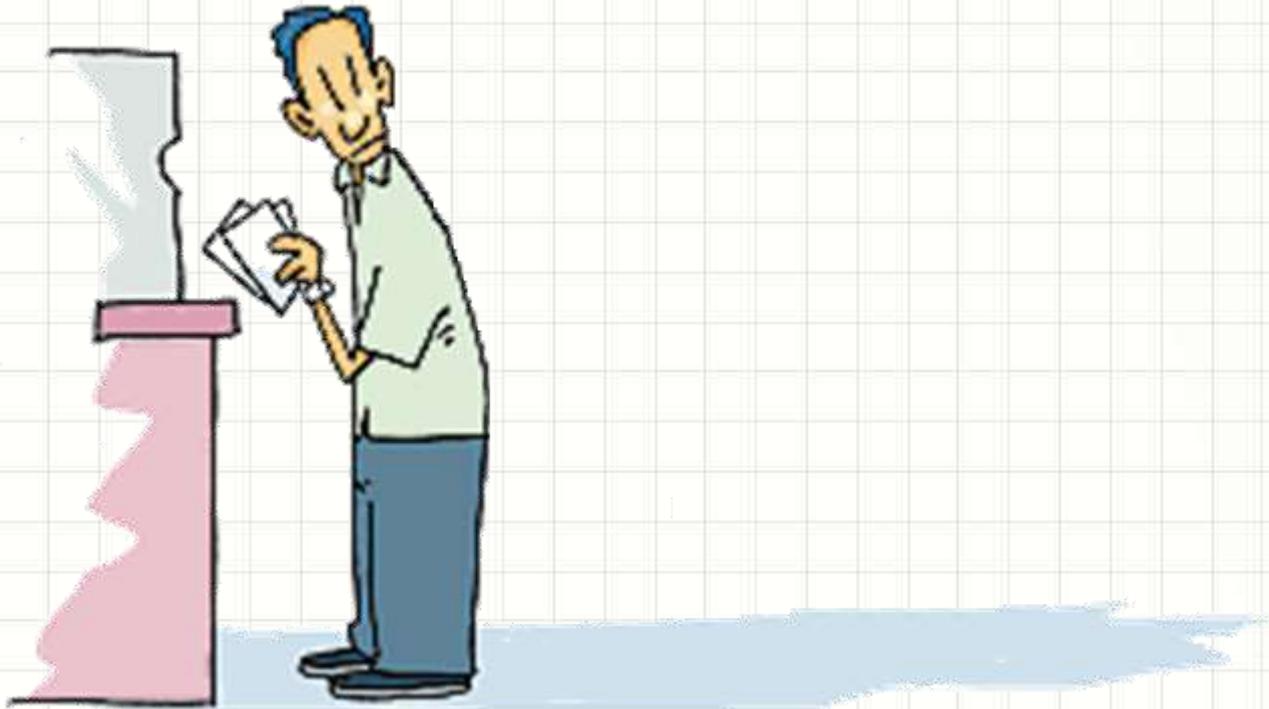
Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



FIFO:
First In
First Out

Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplos: banco



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila de atendimento



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do ba



Fila: 1º a Entrar é 1º a Sair (FIFO)

- É comum: ordem de entrada → saída
- Exemplo: fila do banco



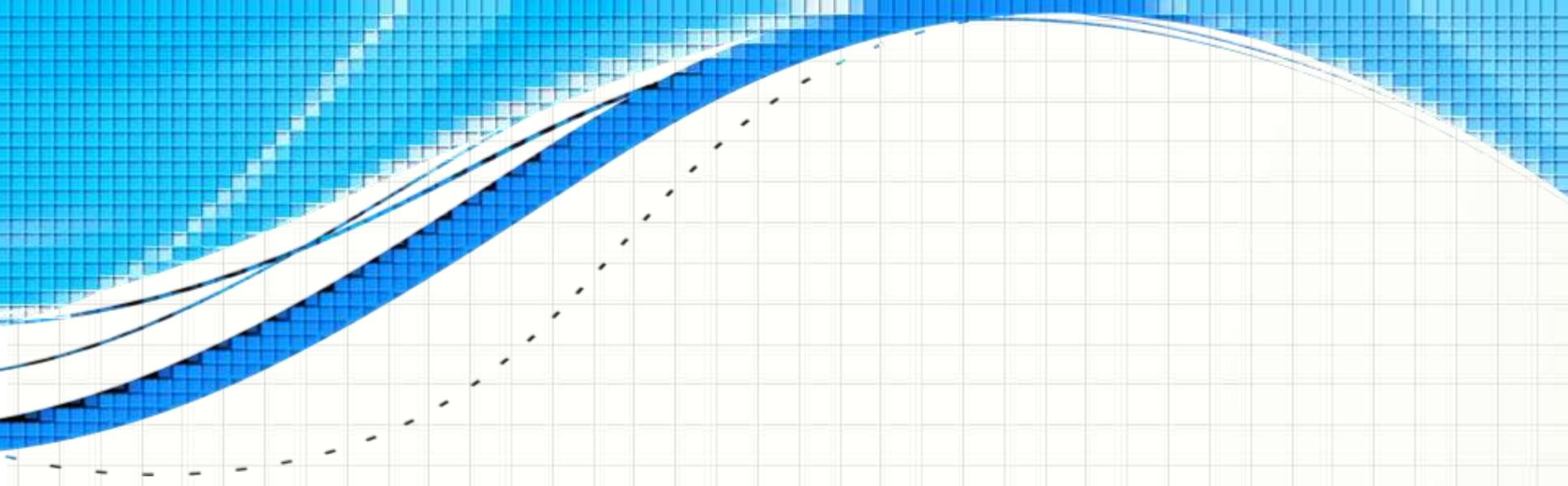
Muito usada em simulações

Implementando Filas

- Comandos `append()` e `pop(0)`
 - Observe o uso:

`aula15ex01.py`

```
# Pessoas chegando na fila e sendo atendidas
fila = []
fila.append("Alberto") # Chegou Alberto
fila.append("Fernanda") # Chegou Fernanda
print( fila.pop(0) ) # Alberto foi atendido
fila.append("Renato") # Chegou Renato
print( fila.pop(0) ) # Fernanda foi atendida
print( fila.pop(0) ) # Renato foi atendido
```



PILHAS:

LISTAS COM OUTRA REGRA PARA ENTRADA E SAÍDA

Pilha: Último a Entrar é 1º a Sair (LIFO)

- Comum: ordem de remoção → armazenamento
- Exemplo: organizando entregas de Pizza



Pilha: Último a Entrar é 1º a Sair (LIFO)

- O que ocorre na Pizzaria?



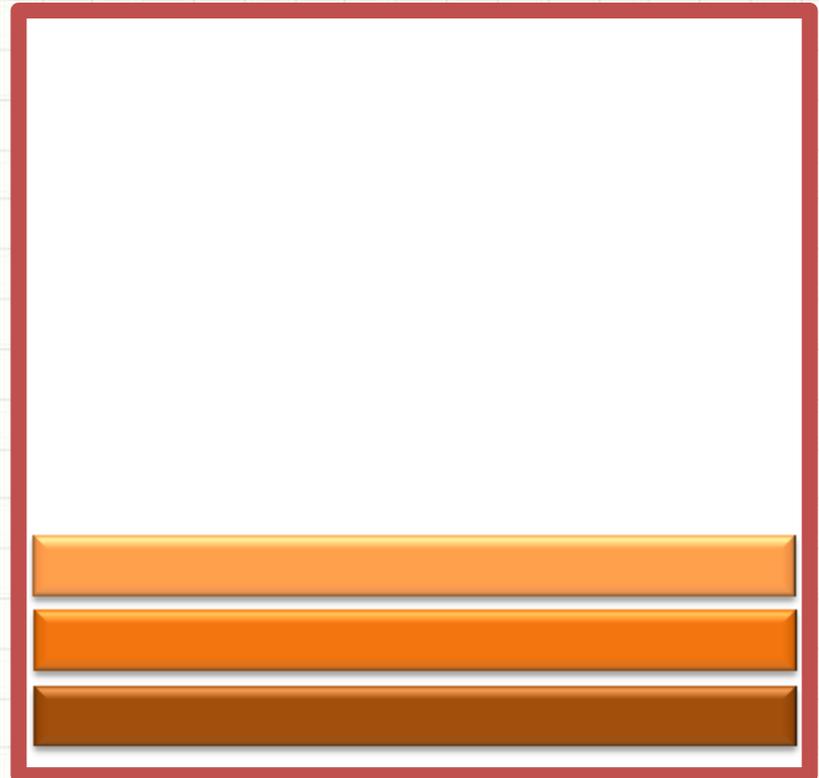
Pilha: Último a Entrar é 1º a Sair (LIFO)

- O que ocorre na Pizzaria?



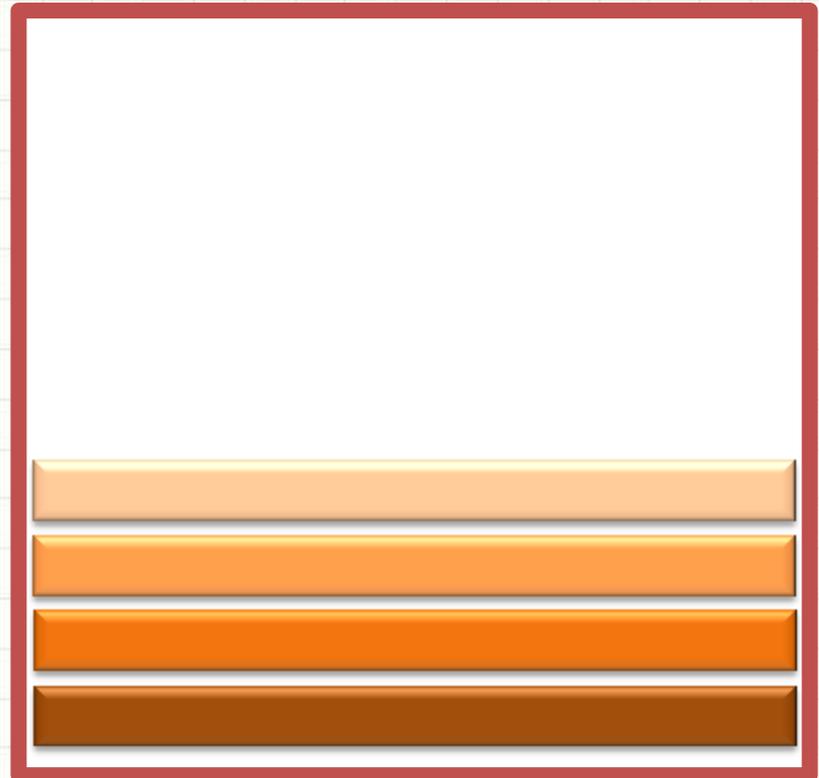
Pilha: Último a Entrar é 1º a Sair (LIFO)

- O que ocorre na Pizzaria?



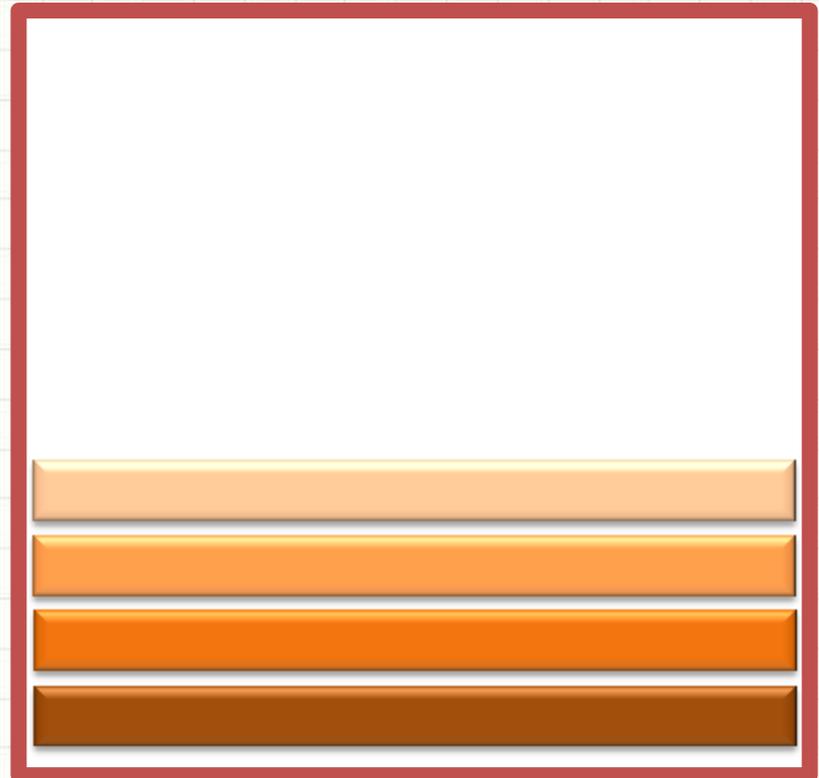
Pilha: Último a Entrar é 1º a Sair (LIFO)

- O que ocorre na Pizzaria?



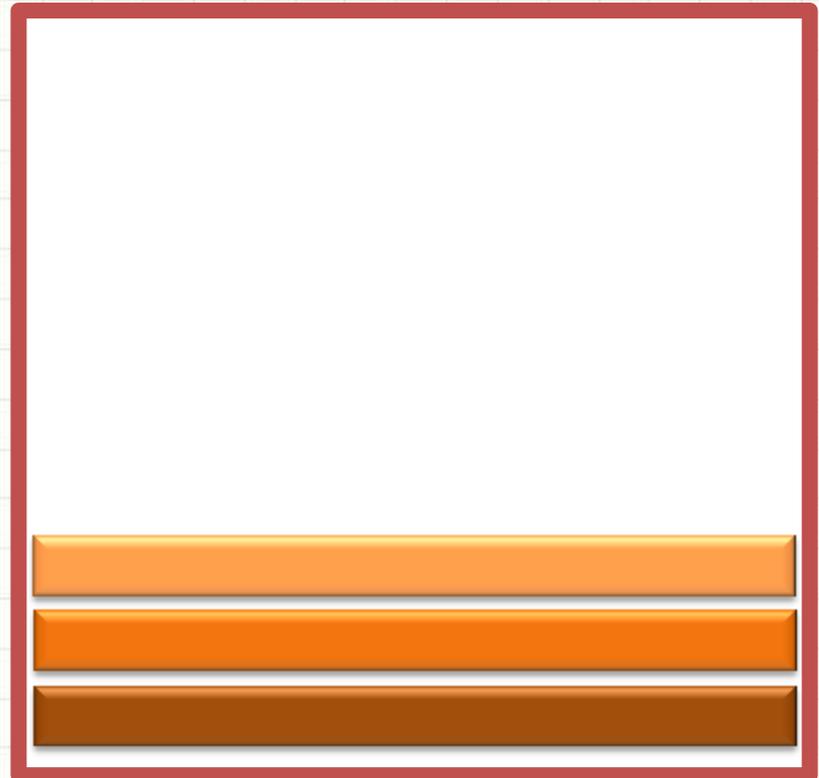
Pilha: Último a Entrar é 1º a Sair (LIFO)

- E nas entregas?



Pilha: Último a Entrar é 1º a Sair (LIFO)

- E nas entregas?



Pilha: Último a Entrar é 1º a Sair (LIFO)

- E nas entregas?



Pilha: Último a Entrar é 1º a Sair (LIFO)

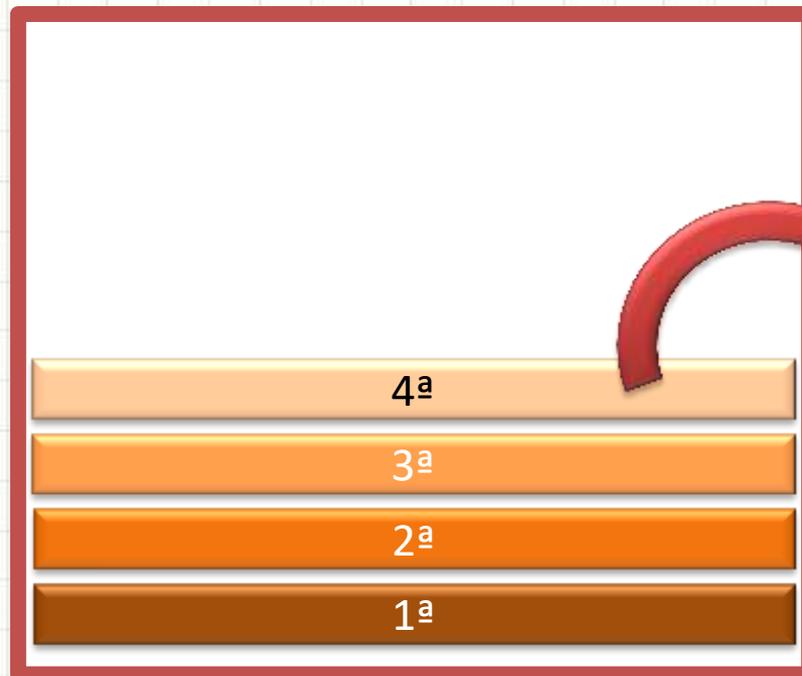
- E nas entregas?



Pilha: Último a Entrar é 1º a Sair (LIFO)

- Observe...
 - A última pizza a entrar...
 - Será a primeira a sair...

LIFO:
Last In
First Out

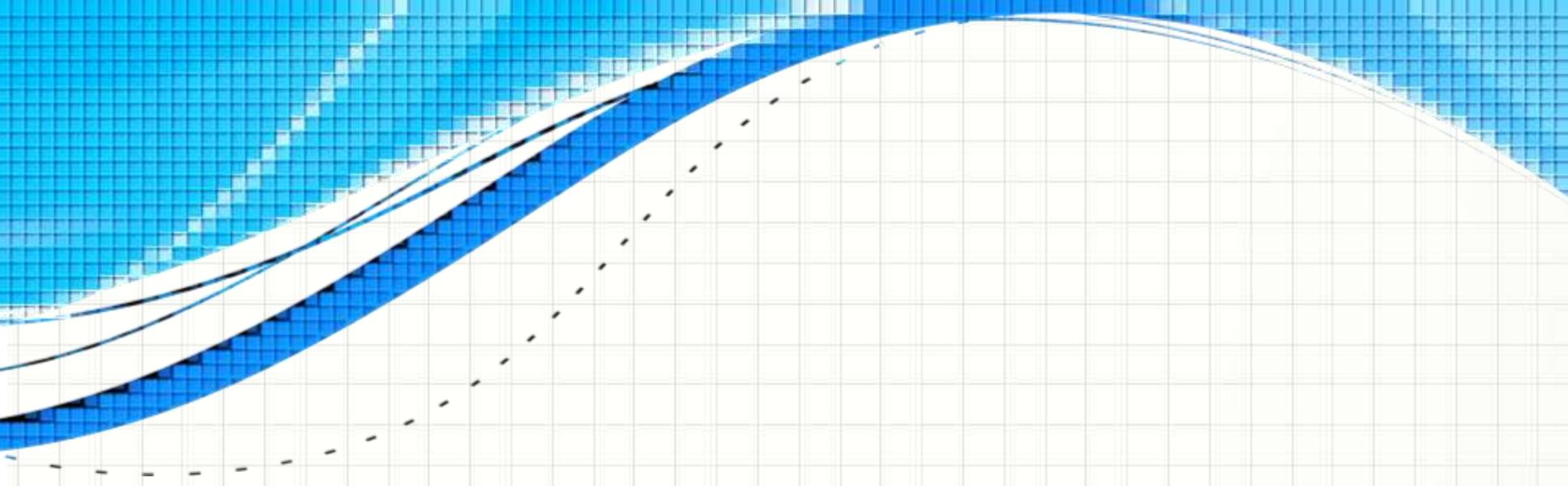


Implementando Pilhas

- Comandos `append()` e `pop(-1)`
 - Observe o uso:

`aula15ex02.py`

```
# Colocando Pizzas na Pilha
pilha = []
pilha.append("Calabresa")
pilha.append("Mussarela")
pilha.append("Pepperoni")
print( pilha.pop() )
print( pilha.pop() )
print( pilha.pop() )
```

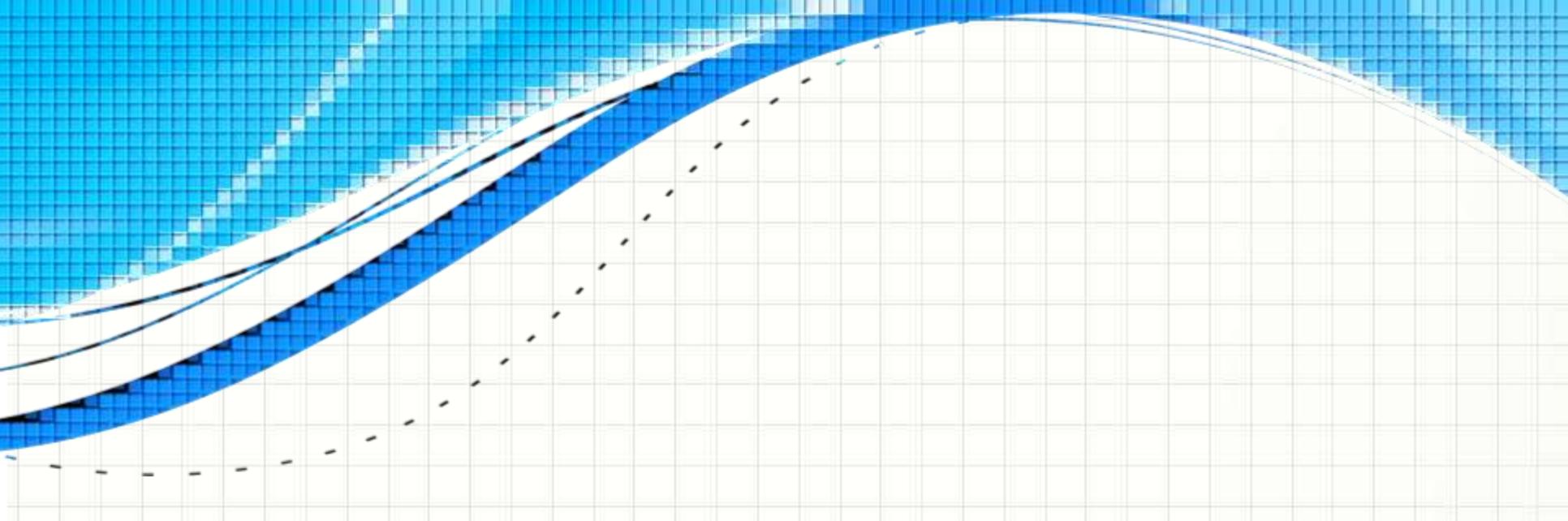


CONCLUSÕES

Resumo

- Listas...
 - Pode guardar muitos elementos!
 - Podemos usar para representar vetores!
 - Agrupam diferentes dados em única variável!
- Pilhas e Filas
- Estruturas e tipos abstratos

SAVA13!



PERGUNTAS?

Exercício para Casa

- Faça um programa que leia 5 valores inteiros em uma lista, pergunte um outro valor para o usuário e, finalmente, indique a primeira posição em que o valor aparece na lista. Se o valor não aparecer, deve imprimir “Valor não aparece na lista!”
- Exemplo: para a lista [1, 10, 7, 28, 45]
- Se o usuário informar o valor 7
- O programa deve apresentar a posição “2”