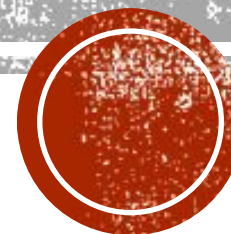


PILHA

Python



PILHA

Características:

- Possui ordem pré-definida de manipulação de elementos;
- As operações são realizadas em apenas uma das extremidades (topo);
- Os primeiros elementos a serem inseridos são os últimos a serem removidos;
- Conceito de LIFO (Last-in-First-out);
- A quantidade de elementos empilhados depende da capacidade de memória;



PILHA

Exemplos:

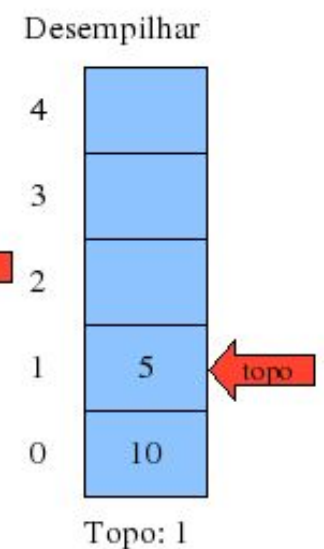
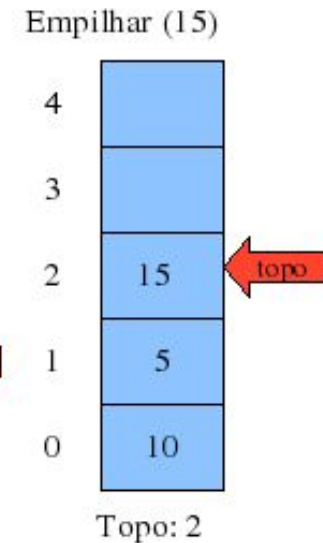
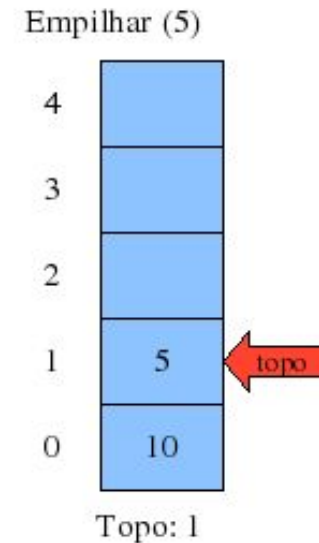
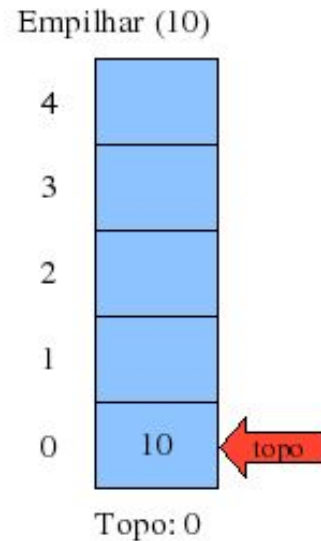
- Funções recursivas em compiladores;
- Mecanismo de desfazer/refazer dos editores de texto;
- Navegação entre páginas Web;



PILHA

Operações:

- Criar a pilha;
- Empilhar;
- Desempilhar;
- Mostrar o topo;
- Verificar se a pilha está vazia;
- Verificar se a pilha está cheia;



PILHA

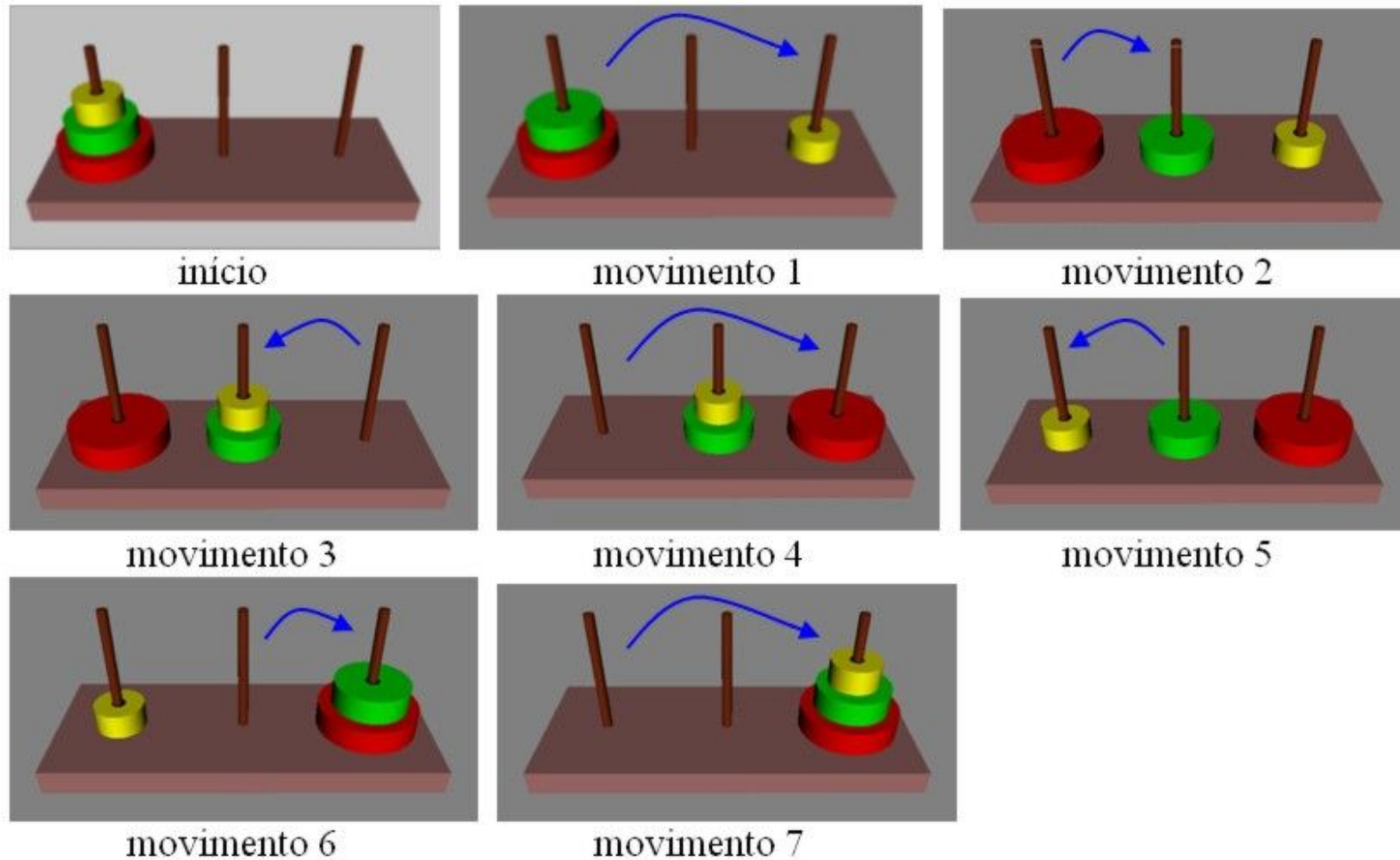
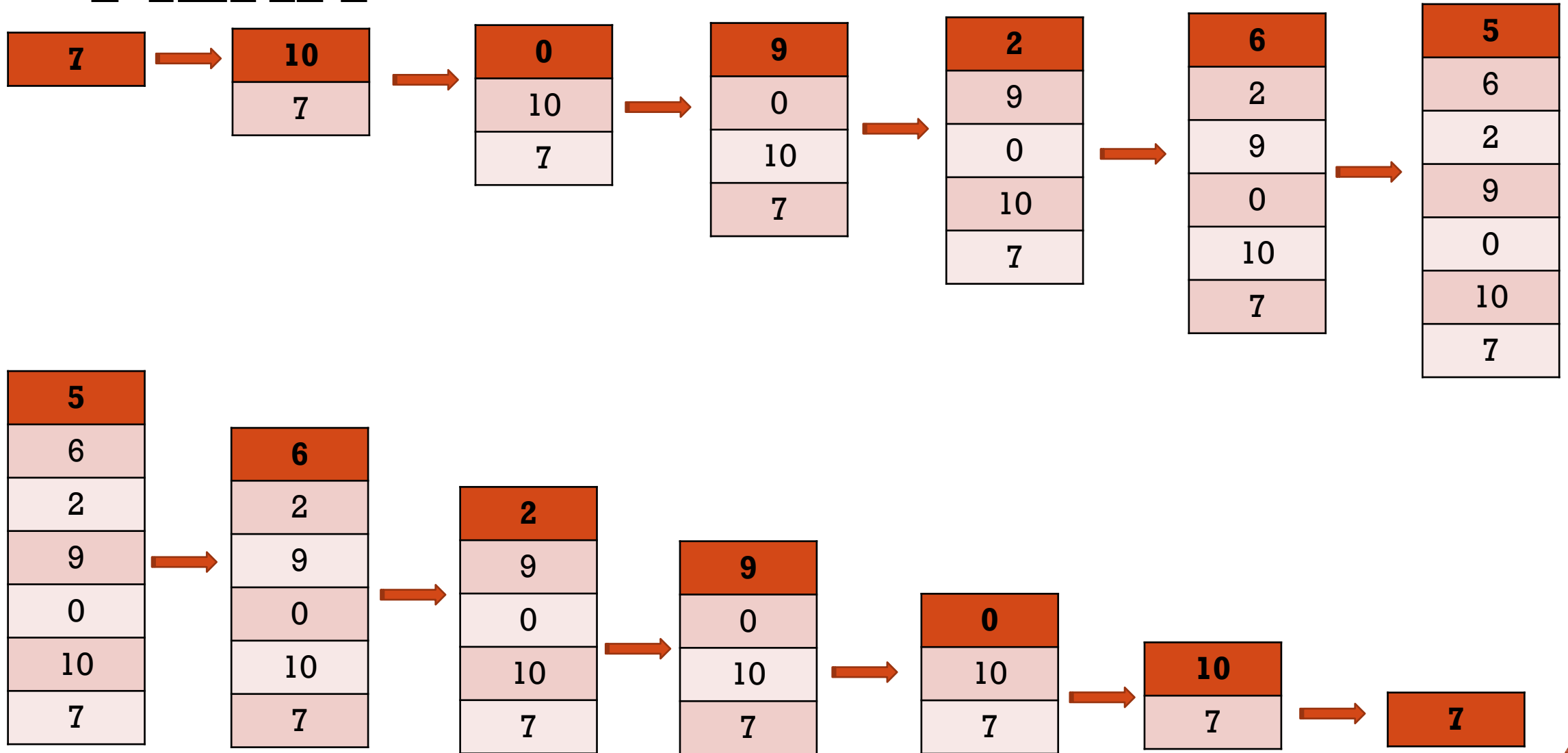


Fig. 5 – Transferência da torre virtual do pino da esquerda para o pino da direita.



PILHA



PILHA - Exemplo

```

pilha = []
print("Informe os valores da pilha")
print()
# PREENCHENDO A PILHA
for cont in range(7):
    pilha.insert(0,int(input('Informe o valor ' + str(cont + 1) + '
para a pilha: ')))
    print(pilha)
# ESVAZIANDO A PILHA
while len(pilha) > 0:
    pilha.pop(0)
    print(pilha)
# FINALIZANDO O PROGRAMA
print("Fim...")
```



PILHA - Exemplo

```
Informe o valor 1 para a pilha: 1  
[1]  
Informe o valor 2 para a pilha: 2  
[2, 1]  
Informe o valor 3 para a pilha: 3  
[3, 2, 1]  
Informe o valor 4 para a pilha: 4  
[4, 3, 2, 1]  
Informe o valor 5 para a pilha: 5  
[5, 4, 3, 2, 1]  
Informe o valor 6 para a pilha: 6  
[6, 5, 4, 3, 2, 1]  
Informe o valor 7 para a pilha: 7  
[7, 6, 5, 4, 3, 2, 1]
```

```
[7, 6, 5, 4, 3, 2, 1]  
[6, 5, 4, 3, 2, 1]  
[5, 4, 3, 2, 1]  
[4, 3, 2, 1]  
[3, 2, 1]  
[2, 1]  
[1]  
[]  
Fim...
```



PILHA - Exemplo

```
pilha = [1, 1, 2, 3, 5]  
print("Pilha: ", pilha)
```

```
pilha.append(8)  
print("Inserindo um elemento: ", pilha)
```

```
pilha.append(13)  
print("Inserindo outro elemento: ", pilha)
```

```
pilha.pop()  
print("Removendo um elemento: ", pilha)
```

```
pilha.pop()  
print("Removendo outro elemento: ", pilha)
```



PILHA - Exemplo

Pilha: [1, 1, 2, 3, 5]

Inserindo um elemento: [1, 1, 2, 3, 5, 8]

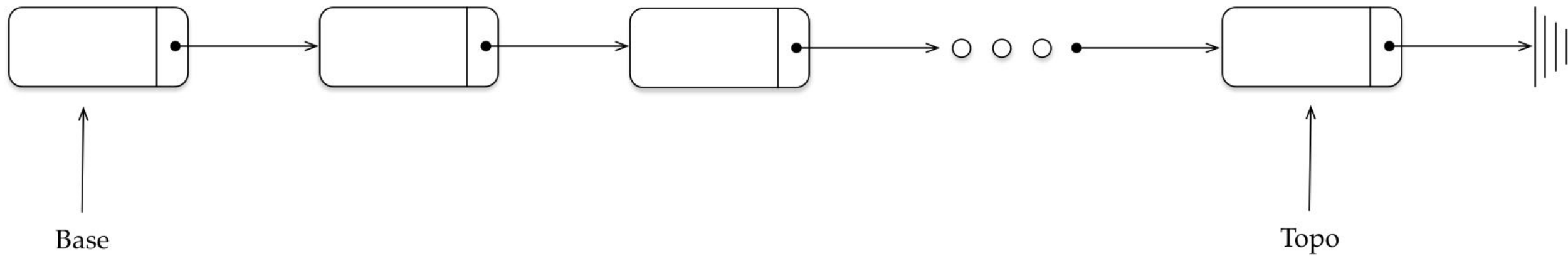
Inserindo outro elemento: [1, 1, 2, 3, 5, 8, 13]

Removendo um elemento: [1, 1, 2, 3, 5, 8]

Removendo outro elemento: [1, 1, 2, 3, 5]



PILHA - Exemplo (Estrutura encadeada)



PILHA - Exemplo

```
class Nodo:
    """Esta classe representa um nodo de uma estrutura encadeada."""
    def __init__(self, dado=0, nodo_anterior=None):
        self.dado = dado
        self.anterior = nodo_anterior

    def __repr__(self):
        return '%s -> %s' % (self.dado, self.anterior)

class Pilha:
    """Esta classe representa uma pilha usando uma estrutura encadeada."""

    def __init__(self):
        self.topo = None

    def __repr__(self):
        return "[" + str(self.topo) + "]"
```



PILHA - Exemplo

```
def insere(self, novo_dado):  
    """Insere um elemento no final da pilha."""  
  
    # Cria um novo nodo com o dado a ser armazenado.  
    novo_nodo = Nodo(novo_dado)  
  
    # Faz com que o novo nodo seja o topo da pilha.  
    novo_nodo.anterior = self.topo  
  
    # Faz com que a cabeça da lista referencie o novo nodo.  
    self.topo = novo_nodo
```



PILHA - Exemplo

```
def remove(self):  
    """Remove o elemento que está no topo da pilha."""  
  
    assert self.topo, "Impossível remover valor de pilha vazia."  
  
    self.topo = self.topo.anterior
```



PILHA - Exemplo

```
# Cria uma pilha vazia.
pilha = Pilha()
print("Pilha vazia: ", pilha)

# Insere elementos na pilha.
for i in range(5):
    pilha.insere(i)
    print("Insere o valor {0} no topo da pilha: {1}".format(i, pilha))

# Remove elementos na pilha.
while pilha.topo != None:
    pilha.remove()
    print("Removendo elemento que está no topo da pilha: ", pilha)
```



PILHA - Exemplo

Pilha vazia: [None]

Inserir o valor 0 no topo da pilha: [0 -> None]

Inserir o valor 1 no topo da pilha: [1 -> 0 -> None]

Inserir o valor 2 no topo da pilha: [2 -> 1 -> 0 -> None]

Inserir o valor 3 no topo da pilha: [3 -> 2 -> 1 -> 0 -> None]

Inserir o valor 4 no topo da pilha: [4 -> 3 -> 2 -> 1 -> 0 -> None]

Removendo elemento que está no topo da pilha: [3 -> 2 -> 1 -> 0 -> None]

Removendo elemento que está no topo da pilha: [2 -> 1 -> 0 -> None]

Removendo elemento que está no topo da pilha: [1 -> 0 -> None]

Removendo elemento que está no topo da pilha: [0 -> None]

Removendo elemento que está no topo da pilha: [None]



PILHA

Referências:

<https://www.cos.ufrj.br/~rfarias/cos121/pilhas.html>

<https://osprogramadores.com/blog/2017/09/10/estruturas-dados-pilha/>

<https://algoritnosempython.com.br/cursos/algoritmos-python/estruturas-dados/pilhas>

https://panda.ime.usp.br/pythonds/static/pythonds_pt/03-EDBasicos/05-PilhaImplementacao.html

<https://www.youtube.com/watch?v=YETRHzwrrvo>



PILHA

Referências:

<https://algoritmoempython.com.br/cursos/algoritmos-python/estruturas-dados/pilhas/#:~:text=Pilhas%20s%C3%A3o%20estruturas%20de%20dados,s%C3%A3o%20geralmente%20implementadas%20com%20arranjos.>

