

# Listas em Objetos

## Sobrecarga de Operadores

CIÊNCIA DA  
COMPUTAÇÃO

ATITUS  
EDUCAÇÃO



ATITUS  
EDUCAÇÃO



- Aula 05 -  
Organização e Abstração

Prof. Me. Lucas R. C. Pessutto

# Problema: Game Score

RANK	SCORE	NAME
1ST	012000	AKI
2ND	009000	CHI
3RD	008000	SEI
4TH	005400	NAO
5TH	003200	IYO
NINTENDO		CREDIT 0

## Passo 1: Classe ItemPlacar

- \* Implemente uma classe para representar um Item do placar.
- \* Cada item possui o nome do jogador e o placar obtido
- \* Faça o método `__str__` que retorna a linha do placar formatada
- \* Utilize encapsulamento!

# Passo 1: Classe ItemPlacar

```
class ItemPlacar:
    def __init__(self, nome, placar):
        self._nome = nome
        self._placar = placar

    @property
    def nome(self):
        return self._nome

    @nome.setter
    def nome(self, valor):
        self._nome = valor

    @property
    def placar(self):
        return self._placar

    @placar.setter
    def placar(self, valor):
        self._placar = valor

    def __str__(self):
        return f"{self._nome:10}\t{self._placar:5}"
```

## Passo 2: Classe Placar

- \* Na classe Placar recebemos como parâmetro o tamanho máximo do placar (Utilize como valor padrão 5).
- \* No construtor dessa classe, crie uma lista vazia para armazenar os itens de score
- \* Defina o método `__str__` dessa classe, que retorna todo o placar
- \* Pense em como fazer a implementação do método para inserir um item no placar:

```
def inserir_item(self, item: ItemPlacar):
```

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```


# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("AAA", 123)`


# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

ItemPlacar("AAA", 123)



# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("BBB", 234)`

<code>ItemPlacar("AAA", 123)</code>

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("BBB", 234)`

<code>ItemPlacar("AAA", 123)</code>

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

ItemPlacar("BBB", 234)
ItemPlacar("AAA", 123)

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("CCC", 200)`

<code>ItemPlacar("BBB", 234)</code>
<code>ItemPlacar("AAA", 123)</code>

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("CCC", 200)`

<code>ItemPlacar("BBB", 234)</code>
<code>ItemPlacar("AAA", 123)</code>

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("DDD", 100)`

<code>ItemPlacar("BBB", 234)</code>
<code>ItemPlacar("CCC", 200)</code>
<code>ItemPlacar("AAA", 123)</code>

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)
ItemPlacar("DDD", 100)



# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("EEE", 400)`

<code>ItemPlacar("BBB", 234)</code>
<code>ItemPlacar("CCC", 200)</code>
<code>ItemPlacar("AAA", 123)</code>
<code>ItemPlacar("DDD", 100)</code>

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("EEE", 400)`

<code>ItemPlacar("BBB", 234)</code>
<code>ItemPlacar("CCC", 200)</code>
<code>ItemPlacar("AAA", 123)</code>

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("EEE", 400)`

<code>ItemPlacar("BBB", 234)</code>
<code>ItemPlacar("CCC", 200)</code>
<code>ItemPlacar("AAA", 123)</code>

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("EEE", 400)`

<code>ItemPlacar("BBB", 234)</code>
<code>ItemPlacar("CCC", 200)</code>
<code>ItemPlacar("AAA", 123)</code>

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("EEE", 400)`

<code>ItemPlacar("BBB", 234)</code>
<code>ItemPlacar("CCC", 200)</code>
<code>ItemPlacar("AAA", 123)</code>

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

ItemPlacar("EEE", 400)
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

ItemPlacar("FFF", 99)

ItemPlacar("EEE", 400)
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

ItemPlacar("EEE", 400)
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("AAA", 123)



# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("GGG", 150)`

<code>ItemPlacar("EEE", 400)</code>
<code>ItemPlacar("BBB", 234)</code>
<code>ItemPlacar("CCC", 200)</code>
<code>ItemPlacar("AAA", 123)</code>

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

`ItemPlacar("GGG", 150)`

<code>ItemPlacar("EEE", 400)</code>
<code>ItemPlacar("BBB", 234)</code>
<code>ItemPlacar("CCC", 200)</code>

# Método Inserir

```
placar = Placar(4)
placar.inserir_item(ItemPlacar("AAA", 123))
placar.inserir_item(ItemPlacar("BBB", 234))
placar.inserir_item(ItemPlacar("CCC", 200))
placar.inserir_item(ItemPlacar("DDD", 100))
placar.inserir_item(ItemPlacar("EEE", 400))
placar.inserir_item(ItemPlacar("FFF", 99))
placar.inserir_item(ItemPlacar("GGG", 150))
```

ItemPlacar("EEE", 400)
ItemPlacar("BBB", 234)
ItemPlacar("CCC", 200)
ItemPlacar("GGG", 150)

# Método Inserir

- \* Seria mais fácil implementar esse método utilizando as funções de lista!

```
def inserir_item(self, item: ItemPlacar):  
    self._scores.append(item)  
    self._scores.sort(reverse=True)  
    if len(self._scores) > self._max_items:  
        self._scores = self._scores[:self._max_items]
```

- \* Porém, o método `sort` não sabe como comparar dois `ItemPlacar`!

# Método Inserir

- \* O método sort utiliza os operadores de comparação (>, <, ==, !=, etc.) para comparar objetos
- \* Podemos dizer como esses operadores se comportam na nossa classe **sobrescrevendo** certas funções mágicas:

$a < b$	<code>a.__lt__(b)</code>
$a \leq b$	<code>a.__le__(b)</code>
$a > b$	<code>a.__gt__(b)</code>
$a \geq b$	<code>a.__ge__(b)</code>
$a == b$	<code>a.__eq__(b)</code>
$a != b$	<code>a.__ne__(b)</code>

# Método Inserir

```
def __lt__(self, outroItem):  
    return self._placar < outroItem._placar
```

```
def __gt__(self, outroItem):  
    return self._placar > outroItem._placar
```

```
def __eq__(self, outroItem):  
    return self._placar == outroItem._placar
```

```
def __le__(self, outroItem):  
    return self._placar <= outroItem._placar
```

```
def __ge__(self, outroItem):  
    return self._placar >= outroItem._placar
```

```
def __ne__(self, outroItem):  
    return self._placar != outroItem._placar
```

```
p1 = ItemPlacar("AAA", 123)
```

```
p2 = ItemPlacar("ZZZ", 123)
```

```
print(p1 == p2)
```

```
print(p1 > p2)
```

```
print(p1 <= p2)
```

# Sobrecarga de Operadores

- \* Podemos fazer nossa classe Placar se comportar como uma lista sobrescrevendo alguns métodos

<code>v in a</code>	<code>a.__contains__(v)</code>
<code>a[k]</code>	<code>a.__getitem__(k)</code>
<code>a[k] = v</code>	<code>a.__setitem__(k,v)</code>
<code>del a[k]</code>	<code>a.__delitem__(k)</code>
<code>len(a)</code>	<code>a.__len__()</code>

# Outros Métodos

\* Operadores Matemáticos:

$a + b$	<code>a.__add__(b);</code>	alternatively <code>b.__radd__(a)</code>
$a - b$	<code>a.__sub__(b);</code>	alternatively <code>b.__rsub__(a)</code>
$a * b$	<code>a.__mul__(b);</code>	alternatively <code>b.__rmul__(a)</code>
$a / b$	<code>a.__truediv__(b);</code>	alternatively <code>b.__rtruediv__(a)</code>
$a // b$	<code>a.__floordiv__(b);</code>	alternatively <code>b.__rfloordiv__(a)</code>
$a \% b$	<code>a.__mod__(b);</code>	alternatively <code>b.__rmod__(a)</code>
$a ** b$	<code>a.__pow__(b);</code>	alternatively <code>b.__rpow__(a)</code>



# Outros Métodos

- \* Conversões para tipos básicos do Python:

<code>bool(a)</code>	<code>a.__bool__()</code>
<code>float(a)</code>	<code>a.__float__()</code>
<code>int(a)</code>	<code>a.__int__()</code>

# Sobrecarga de Operadores

- \* A sobrecarga de operadores costuma ser utilizada em três ocasiões especiais:
  - Para emular tipos numéricos que não existem no Python
  - Converter um objeto para um tipo básico do Python (str, int, etc.)
  - Implementar coleções