

# Plano de recuperação

## Módulo 4: Estrutura de dados estáticas

Para cada um dos programas seguintes, deverá transcrever para o papel o código, comentando todas as linhas.

**Deverá ainda fazer uma traçagem do código e explicar, na generalidade, o que faz o programa.**

No final deverá combinar com o professor a realização de um exercício de validação dos conhecimentos.

---

### Código 1

```
L = [(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')]
L = [t for t in L if t]
print(L)
```

---

### Código 2

```
t = (100, 200, 300)
print('This is a tuple {0}'.format(t))
```

---

### Código 3

```
num = [10,20,30,(10,20),40]
ctr = 0
for n in num:
    if isinstance(n, tuple):
        break
    ctr += 1
print(ctr)
```

---

### Código 4

```
def average_tuple(nums):
    result = [sum(x) / len(x) for x in zip(*nums)]
    return result

nums = ((10, 10, 10, 12), (30, 45, 56, 45), (81, 80, 39, 32), (1, 2, 3, 4))
print ("Tuplo original: ")
```

```

print(nums)
print("\nMédia:\n",average_tuple(nums))

nums = ((1, 1, -5), (30, -15, 56), (81, -60, -39), (-10, 2, 3))
print ("\nTuplo original: ")
print(nums)
print("\nnMédia:\n",average_tuple(nums))

```

---

### Código 5

```

x = (1,2,3,4)
y = (3,5,2,1)
z = (2,2,3,1)
print("Original lists:")
print(x)
print(y)
print(z)
print("\nElement-wise sum of the said tuples:")
result = tuple(map(sum, zip(x, y, z)))
print(result)

```