



#### Unidade 2: Iniciação à programação em Python

#### Lição 3: O ciclo não limitado WHILE

Nesta terceira lição da Unidade 2 vamos descobrir como repetir um procedimento ou um conjunto de instruções utilizando um ciclo não limitado WHILE.

#### Objetivos:

- Explorar e implementar o ciclo não limitado WHILE
- Utilizar o ciclo WHILE em exemplos simples

Por vezes, num programa, é útil e/ou necessário repetir uma ou várias instruções um número, não previamente definido, de vezes. Se o número de repetições do processo não é conhecido, então utilizamos um ciclo não limitado **WHILE**.

A sintaxe de um ciclo **WHILE** é a seguinte:

#### ALGORITMO

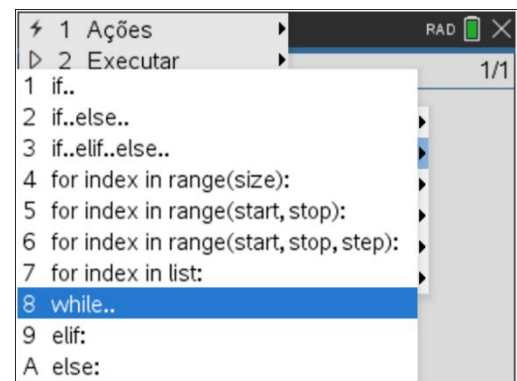
Enquanto a condição for verdadeira:

```
Instrução1
Instrução2
...
```

#### LINGUAGEM PYTHON

while condição

```
Instrução1
Instrução2
...
```



Não existe nenhuma instrução para indicar o fim de ciclo. Faz-se pela indentação, ou seja, o deslocamento para a direita de uma ou mais linhas é que assinala o fim do ciclo.

#### EXEMPLO:

Termos de uma sucessão definida por recorrência.

Elabora um programa que permita determinar qual a ordem,  $n$ , do maior termo inferior a 1 da sucessão definida por recorrência por:

$$\begin{cases} c_0 = 3.4 \\ c_{n+1} = 0.8 * c_n \end{cases}$$

#### ALGORITMO

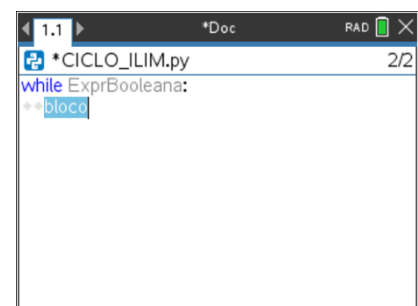
Procedimentos:

```
n ← 0
c ← 3.4
Enquanto c ≥ 1
    n ← n + 1
    c ← 0.8 * c
FimEnquanto
```

#### IMPLEMENTAÇÃO DO ALGORITMO:

Vamos criar um programa para se entender melhor o que é um ciclo, assim como, o que é um processo iterativo.

- Inicie um novo programa em Python e designe-o por “CICLO\_ILIM”.
- A função **While** acede-se clicando na tecla **[menu]** e selecione submenu **4: Planos integrados**, depois **2: Controlo**, e por fim a opção **8: while..**, desta forma surgirá já a sua estrutura com os campos para preencher.





- Enquanto a variável  $c$ , termo da sucessão, for superior ou igual ao limite, a variável  $n$ , ordem do termo, será incrementada em 1.
- Para tal, defina uma função em Python de forma que entremos dois valores, termo inicial e limite, se obtenha o número de termos inferiores ao valor do limite.
- Pressione simultaneamente as teclas **ctrl** e **B** para verificar a sintaxe e guardar o programa.
- Execute o programa, clicando simultaneamente nas teclas **ctrl** e **R**. Abrir-se-á uma nova página com o interpretador de Python (Shell) onde foi executado o programa.
- No Shell pode agora usar a função **a** definida no programa CICLO\_ILIM.
- Determina, pela função **a**, a resposta ao problema, isto é, **a(3,4,1)**.

```

1.1 | *U2C3 | RAD | X
CICLO_ILIM.py guardado com sucesso
def a(c,limite):
    n=0
    while c>=limite:
        n=n+1
        c=0.8*c
    return n

```

```

1.1 | 1.2 | *U2C3 | RAD | X
Shell Python | 5/5
>>>#Running CICLO_ILIM.py
>>>from CICLO_ILIM import *
>>>a(3,4,1)
6
>>>

```

### APLICAÇÃO DAS APRENDIZAGENS:

#### Os ressaltos da bola saltitona

Uma bola é largada no ar a uma altura de 1,20 metros e ressalta no solo atingindo uma altura de 40% da altura do ressalto anterior.

Elabore um algoritmo que permita obter o número de ressaltos realizados até a altura atingida pela bola ser estritamente inferior a 1 cm.

Baseie-se nos seguintes passos:

- Defina uma variável  $h$  na qual se guarde a altura do ressalto, em cm, e um variável  $r$  para guardar o número de ressaltos.
- Terá que ser repetido o cálculo da altura, " $h$  toma o valor  $0.40 \times h$ ", várias vezes sem se saiba antecipadamente o número de repetições.
- Após cada cálculo será testada a condição  $h \geq 1$ , sendo que o ciclo de repetição dos cálculos será executado enquanto a condição for verdadeira.



#### ALGORITMO

##### Procedimentos:

```

h ← 120
r ← 0
Enquanto h ≥ 1
    r ← r + 1
    h ← 0.4 * h
FimEnquanto
Escrever r

```

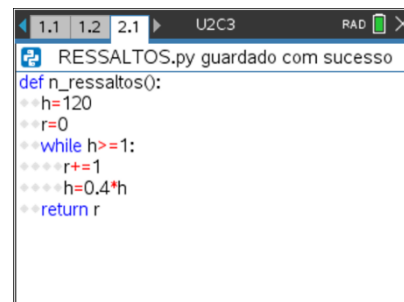


### IMPLEMENTAÇÃO DO ALGORITMO:

Vamos, agora, construir o programa na linguagem Python, usando o editor da aplicação TI-Python.

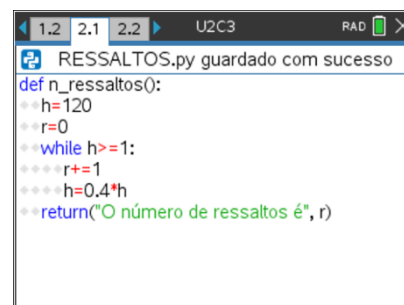
- Inicie um novo programa em Python e designe-o por “RESSALTOS”.
- Propõe-se que use uma função. Teremos, assim, o cuidado de respeitar a indentação (um para o ciclo **While** e um segundo para a função **return**) e obteremos no fim o valor da variável **r**.
- Construa a função **n\_ressaltos**, desta vez será uma função sem argumentos, começando por atribuir os valores iniciais às variáveis **h** e **r**.
- De seguida coloque o ciclo não limitado **While**, colocando a condição e as respetivas linhas de instrução.
- Pressione simultaneamente as teclas **ctrl** e **B** para verificar a sintaxe e guardar o programa.
- Execute o programa, clicando simultaneamente nas teclas **ctrl** e **R**. Depois, na página do interpretador, escreva o nome da função ou clique na tecla **var** e selecione a função e prima **enter**.
- Enriqueça o programa acrescentando uma mensagem, por exemplo: “*O número de ressaltos é* “. Para tal, no editor acrescenta à função **return** o texto pretendido, isto é, escreva:

•• `return("O número de ressaltos é ", r)`



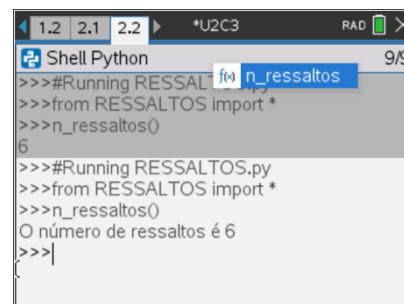
```

1.1 1.2 2.1 U2C3 RAD
RESSALTOS.py guardado com sucesso
def n_ressaltos():
    h=120
    r=0
    while h>=1:
        r+=1
        h=0.4*h
    return r
    
```



```

1.2 2.1 2.2 U2C3 RAD
RESSALTOS.py guardado com sucesso
def n_ressaltos():
    h=120
    r=0
    while h>=1:
        r+=1
        h=0.4*h
    return("O número de ressaltos é", r)
    
```



```

1.2 2.1 2.2 *U2C3 RAD
Shell Python 9/9
>>>#Running RESSALTOS.py
>>>from RESSALTOS import *
>>>n_ressaltos()
6
>>>#Running RESSALTOS.py
>>>from RESSALTOS import *
>>>n_ressaltos()
O número de ressaltos é 6
>>>|
    
```

### ATENÇÃO:

Pode ser necessário manter os valores intermédios dos cálculos, para por exemplo serem reutilizados ou simplesmente ficarem guardados (termos de uma sequência numérica, por exemplo). Nesse caso será recomendável a utilização de listas, sendo no exemplo anterior o valor pretendido o último elemento dessa lista.

### UM DESAFIO:

Enriquecer o programa anterior, RESSALTOS, de forma a obter-se a distância total percorrida pela bola até ela parar, assumindo que os ressaltos da bola apenas registam deslocamentos verticais.