

Ciclos Definidos e Indefinidos

UNIDAD 5 Parte I



Introducción a la Programación (11071)
Departamento de Ciencias Básicas
Universidad Nacional de Luján



ORGANIZACIÓN

En esta clase vamos a hablar de los siguientes puntos:

- ★ Introducción
- ★ Ciclos definidos
- ★ Sintaxis de ciclos definidos
- ★ Entendiendo range
- ★ Ciclos Indefinidos
- ★ Sintaxis de ciclos indefinidos
- ★ Notas adicionales



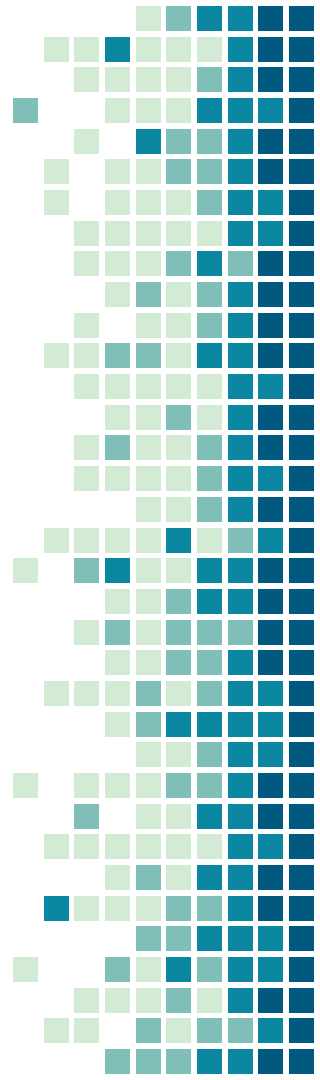
INTRODUCCIÓN

La programación estructurada es un paradigma que se basa en el uso de subrutinas (funciones) y tres estructuras básicas:

- La **secuencia** (bloques de código)
- La **selección** (condicionales, decisión o "ifs")
- La **iteración** (ciclos, bucles o loops).

Si bien podríamos resolver la mayoría de los problemas sin utilizar ciclos esto resultaría tedioso.

Por ejemplo: desarrollar un programa que sume 100 números.



INTRODUCCIÓN

El concepto principal que debemos entender al referirse a un ciclo es la posibilidad de repetir la ejecución sucesiva de un mismo bloque de código.

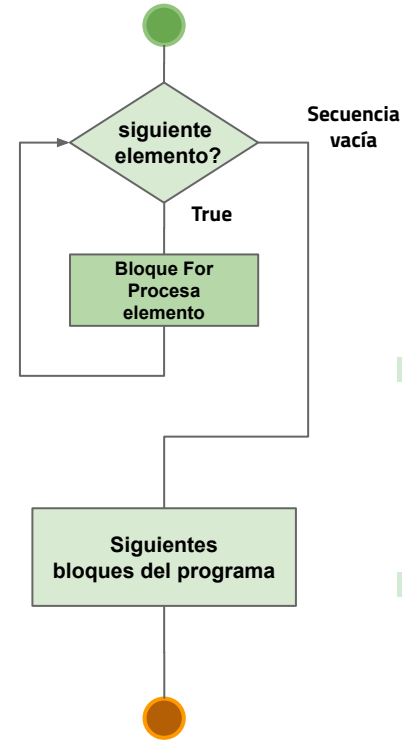
Dentro de los ciclos encontramos dos tipos:

- ★ **Definidos:** Aquí, se conoce previamente la cantidad de veces que se ejecutará el ciclo.
- ★ **Indefinidos:** Este tipo de ciclos posee una condición y el ciclo se sigue ejecutando mientras la condición se cumpla.

CICLOS DEFINIDOS

Los *ciclos definidos* tienen la característica principal que **se conoce previamente** la cantidad de iteraciones que serán ejecutadas.

Por ejemplo, si se necesitara ingresar las notas de 10 alumnos y calcular su promedio.



SINTAXIS DE CICLOS DEFINIDOS

Python implementa **ciclos definidos** con la estructura *for*, la cual es de la forma:

```
for <variable_interna> in <iterable>:  
    <cuerpo>
```

donde, ***variable_interna*** es el nombre con el cual nombramos a cada elemento de la secuencia que llamamos *iterable* y el bloque de código a ejecutar lo llamamos *cuerpo*, *bloque interno*, etc.

Iterable es una característica de los tipos de variable que permite ser usada como secuencia.

SINTAXIS DE CICLOS DEFINIDOS

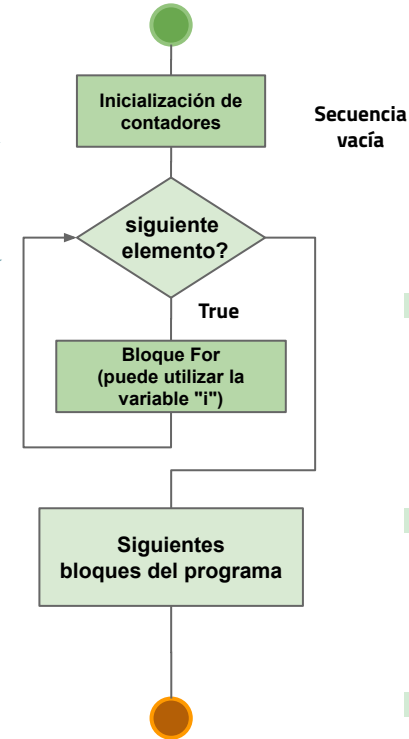
Resolución del ejemplo

Se requiere ingresar las notas de 10 alumnos y calcular su promedio.

```
suma = 0
cantidad = 10

for i in range(cantidad):
    nota = input("Ingrese la nota del alumno ")
    suma += int(nota)

promedio = suma/cantidad
print("EL promedio de calificaciones es", promedio)
```



SINTAXIS DE CICLOS DEFINIDOS

Algunos ejemplos de iterables son *str* o *list*.

Ahora pensemos en imprimir los números del 1 al 100.

El programador debería escribir una lista de 100 elementos?



```
for letra in "Hola":  
    print(letra)
```

```
> H  
> o  
> l  
> a
```

```
for nro in [1, 2, 3]:  
    print(nro)
```

```
> 1  
> 2  
> 3
```


SINTAXIS DE CICLOS DEFINIDOS

En otros lenguajes como Pascal, Javascript, C, C++ entre muchos otros, la sintaxis es un poco distinta.

La variable de control se denomina normalmente con la letra '*i*', '*j*' o '*k*' y no debe ser alterada por ningún motivo.

```
/* C for loop */  
for (i = 0; i < 5; i++) {  
    some_function(i);  
}
```

```
{ Pascal for loop }  
for i := 1 to 5 do  
begin  
    (*statement*);  
end;
```

```
// Javascript for loop  
for (var i = 0; i < 5; i++) {  
    // ...  
}
```

ENTENDIENDO RANGE

range nos devolverá una secuencia según los parámetros que usemos.

```
# range docstring:  
range(stop) -> range object  
range(start, stop[, step]) -> range object
```

Podemos ejecutar *range* con 1, 2 o 3 parámetros, estos son **start**, **stop** y **step** (paso).

Veamos cómo entender esto.

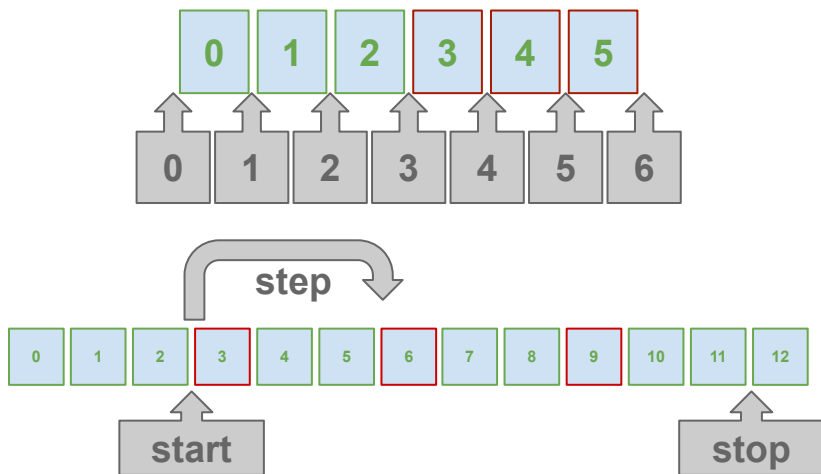
```
for nro in range(3):  
    print(nro)  
  
> 0  
> 1  
> 2
```

```
for nro in range(3, 6):  
    print(nro)  
  
> 3  
> 4  
> 5
```

```
for nro in range(3, 12, 3):  
    print(nro)  
  
> 3  
> 6  
> 9
```

ENTENDIENDO RANGE

range nos devolverá una secuencia según los parámetros que usemos.



```
for nro in range(3):  
    print(nro)  
> 0  
> 1  
> 2
```

```
for nro in range(3, 6):  
    print(nro)  
> 3  
> 4  
> 5
```

```
for nro in range(3, 12, 3):  
    print(nro)  
> 3  
> 6  
> 9
```

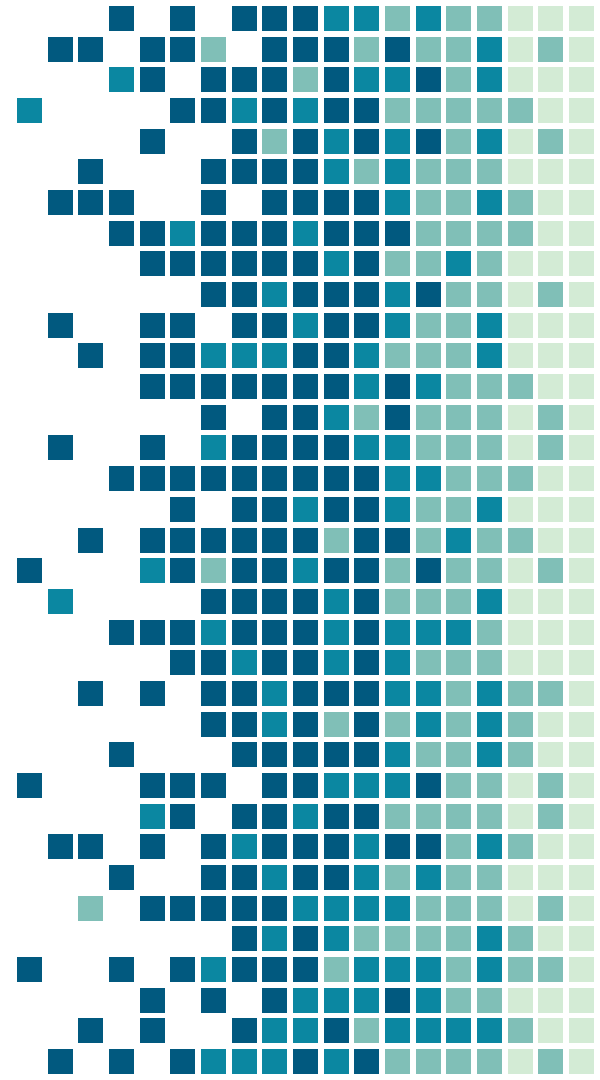


Ciclos Definidos e Indefinidos

UNIDAD 5 Parte II



Introducción a la Programación (11071)
Departamento de Ciencias Básicas
Universidad Nacional de Luján



ORGANIZACIÓN

En esta clase vamos a hablar de los siguientes puntos:

- ★ Introducción
- ★ Ciclos definidos
- ★ Sintaxis de ciclos definidos
- ★ Entendiendo range
- ★ Ciclos Indefinidos
- ★ Sintaxis de ciclos indefinidos
- ★ Notas adicionales



INTRODUCCIÓN

El concepto principal que debemos entender al referirse a un ciclo es la posibilidad de repetir la ejecución sucesiva de un mismo bloque de código.

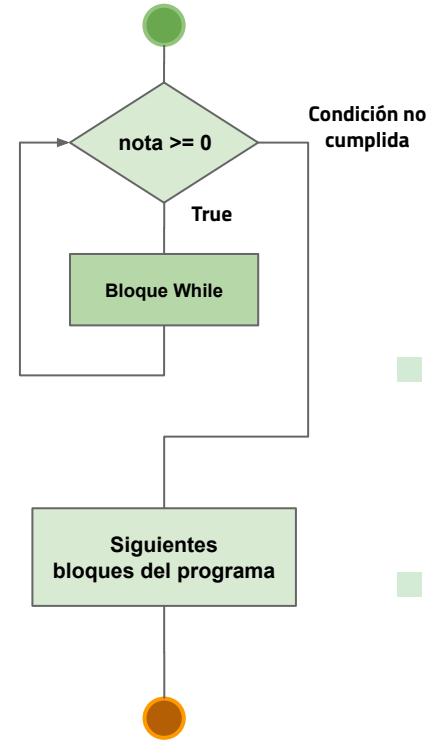
Dentro de los ciclos encontramos dos tipos:

- ★ **Definidos:** Aquí, se conoce previamente la cantidad de veces que se ejecutará el ciclo.
- ★ **Indefinidos:** Este tipo de ciclos posee una condición y el ciclo se sigue ejecutando mientras la condición se cumpla.

CICLOS INDEFINIDOS

En los *ciclos indefinidos*, a diferencia de los ciclos definidos, ***no se conoce previamente la cantidad de iteraciones a realizar.***

Estos ciclos poseen una ***condición*** y el ciclo continúa iterando en el bloque especificado **mientras la condición sea verdadera.**



SINTAXIS EN PYTHON

En Python, los ciclos indefinidos se implementan con la instrucción *while*, la cual posee la siguiente sintaxis:

```
while <expresión>:  
    <cuerpo>
```

donde, ***expresión*** es la **condición** que se evaluará al comienzo de cada iteración y el *cuerpo* es el bloque de código a ejecutar **mientras la condición sea verdadera**.

La *expresión* es igual a las usadas en condicionales y pueden ser compuestas con operadores lógicos.

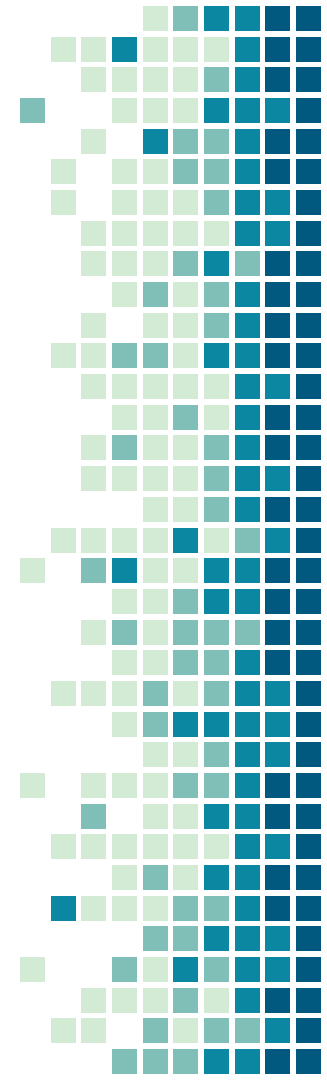
IMPLEMENTACIÓN EN PYTHON

Ejemplo:

Se deben ingresar por teclado las notas de los alumnos en una determinada asignatura, o un 99 si el alumno estuvo ausente.

Preparar un algoritmo para calcular e imprimir la nota promedio, recordando que el alumno ausente no se promedia.

El algoritmo debe terminar cuando se ingrese un valor negativo.



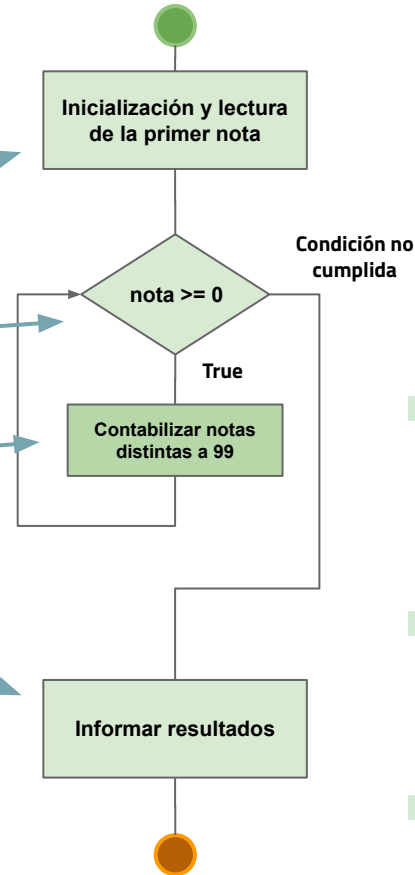
IMPLEMENTACIÓN EN PYTHON

Resolución del ejemplo planteado:

```
suma = 0
cantidad = 0
nota = input("Ingrese la primer nota ")
nota = int(nota)

while nota >= 0:
    if nota <> 99:
        suma += nota
        cantidad += 1
    nota = input("Ingrese la siguiente nota ")
    nota = int(nota)

if cantidad > 0:
    promedio = suma/cantidad
    print("EL promedio de calificaciones es", promedio)
```



NOTAS ADICIONALES

- ★ **Atención!!!** Al utilizar ciclos indefinidos tener cuidado al definir la condición de corte o podemos generar un ciclo infinito y nuestro programa no podrá salir del mismo.
- ★ Los lenguajes de programación, como Python, poseen keywords para alterar el flujo de los ciclos, como *break*, *continue* o usar *return* dentro del cuerpo del ciclo.
Su uso **está totalmente desaconsejado** para un curso de introducción a la programación ya que fomenta malas prácticas de programación.

