



Argentina  
programa  
4.0

**.UBA**fiuba   
FACULTAD DE INGENIERÍA



# PYTHON

## Tuplas

## Características y Manejo

**Lic. Gustavo Bianchi**

Junio 2023

# Contenido

---

- Estructura Interna, composición y organización.
- Cómo acceder a los elementos que la componen.
- Cómo agregar, modificar y eliminar elementos.
- Cómo hacemos para recorrerlas.
- Cómo podemos generarlas.
- Cómo las podemos ordenar.

# ¿Qué son las tuplas?

→ Son una **Colección o agrupación de elementos** en secuencia, encerrados entre paréntesis

`(2, 23, 8, 48, 5, 0)`

→ Pueden ser de distinto tipo

`(235, "azul", 28.3, "rojo")`

→ Son **INMUTABLES**, su contenido **NO puede ser modificado**

# ¿Cómo debemos declararlas?

→ Para declarar una tupla podemos o no encerrar los elementos entre paréntesis; pero sí deben estar separados por una “,”

```
>>> tupla = (2, 5, 23, -18)
```

o

```
>>> tupla = 2, 5, 23, -18
```

Ambos generan la misma tupla  
(2, 5, 23, -18)

→ Si necesitamos declarar una tupla con un único elemento es necesario que seguido al elemento coloquemos una “,”

```
>>> tupla_1_elemento = (5,)
```

# ¿Cómo se accede a los elementos?

→ Cada elemento ocupa una posición comenzando en cero

(2, 23, 8, 48, 5, 0)  
0 1 2 3 4 5

→ Se accede a los elementos a través de la posición que ocupan

```
>>> tupla = (2, 23, 8, 48, 5, 0)
```

```
>>> tupla[3]
```

48

Debemos encerrar la posición entre corchetes

→ También puedo acceder a una posición, desde el final de la tupla

```
>>> tupla[ - 1]
```

0

Accedo al último elemento de la lista

```
>>> tupla[ -4 ]
```

8

Accedo al 4to. elemento desde el final

# ¿Qué operaciones podemos hacer?

→ Se pueden concatenar tuplas y de esta forma obtener una nueva tupla

Ejemplos

```
>>> tupla = tupla + (-1, -10, 23)
```

```
>>> tupla = tupla_1 + tupla_2
```

```
>>> tupla = (51, -100, 3) + (-15, 23) + otra_tupla
```

# Y además también podemos .....

```
>>> tupla = [2, 23, 8, 48, 5, 8, 0]
```

Conocer la cantidad de  
elementos que hay en la tupla

```
>>> len( tupla )  
7
```

Conocer la cantidad de  
ocurrencias de un elemento

```
>>> tupla.count(8)  
2
```

Conocer la posición de un  
elemento

```
>>> tupla.index(23)  
1
```

# ¿Cómo hacemos para recorrerlas?

```
>>> tupla = (2, 8, 23, 48, 5, 8)
```

Si quisiéramos mostrar cada elemento de la tupla en una línea diferente

```
for elemento in tupla:  
    print(elemento)
```

```
2  
8  
23  
48  
5  
8
```

Si quisiéramos mostrar un elemento por línea hasta encontrar el primer elemento impar

```
posicion = 0  
while posicion < len(tupla) and \  
    tupla[posicion] % 2 == 0:  
    print(tupla[posicion])  
    posicion += 1
```

```
2  
8
```

**[www.ingenieria.uba.ar](http://www.ingenieria.uba.ar)**

**f**    /ingenieriauba

 /FIUBAoficial