



Bucles, listas, tuplas, diccionarios

Gutiérrez Cruz Abel Isaías

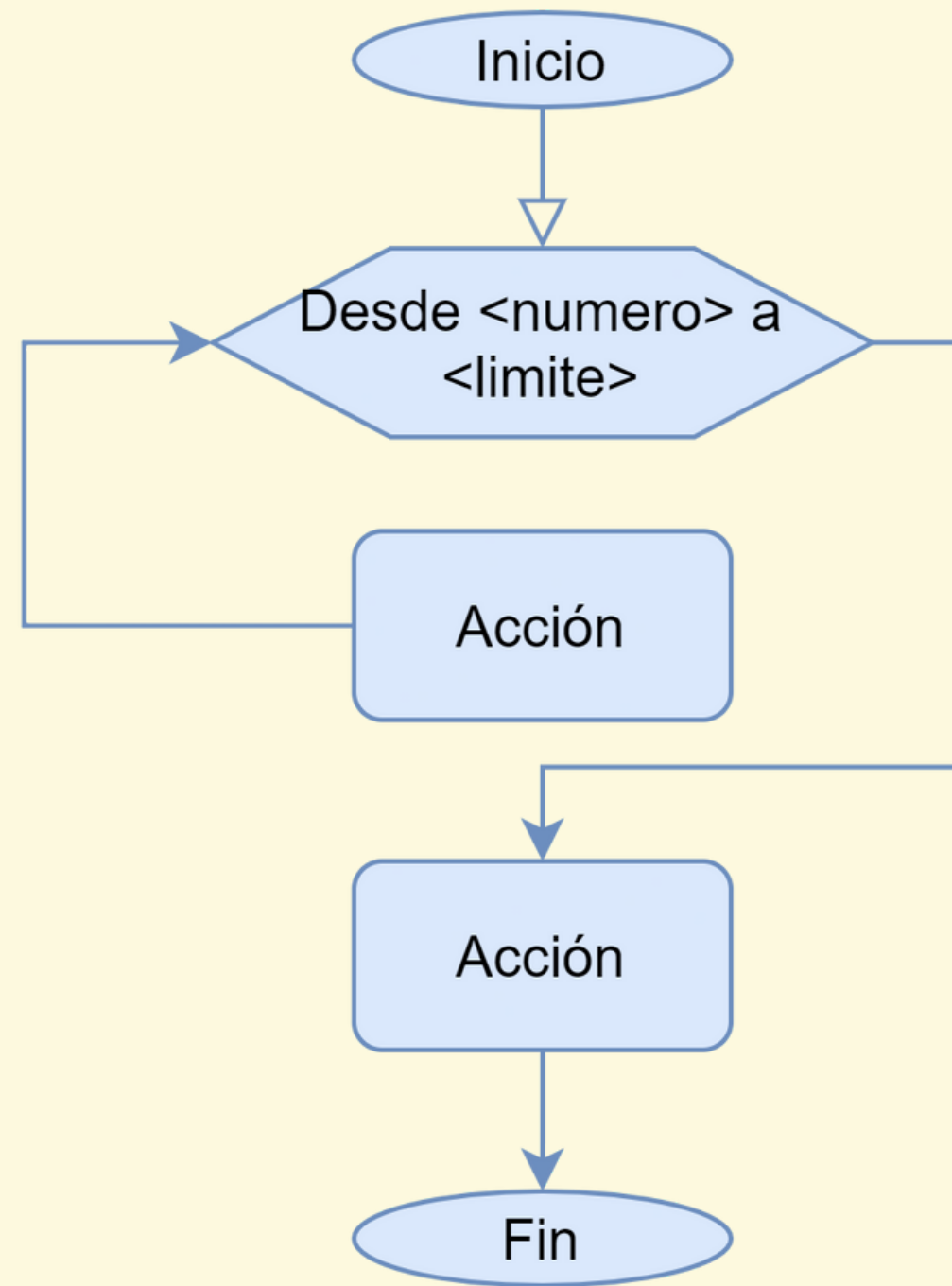
Bucles

Algoritmos que necesitan repetir un conjunto de acciones con las mismas características que se pueden ejecutar una cantidad de veces específica. Estos ciclos pueden ser establecidos por el programador o no

For



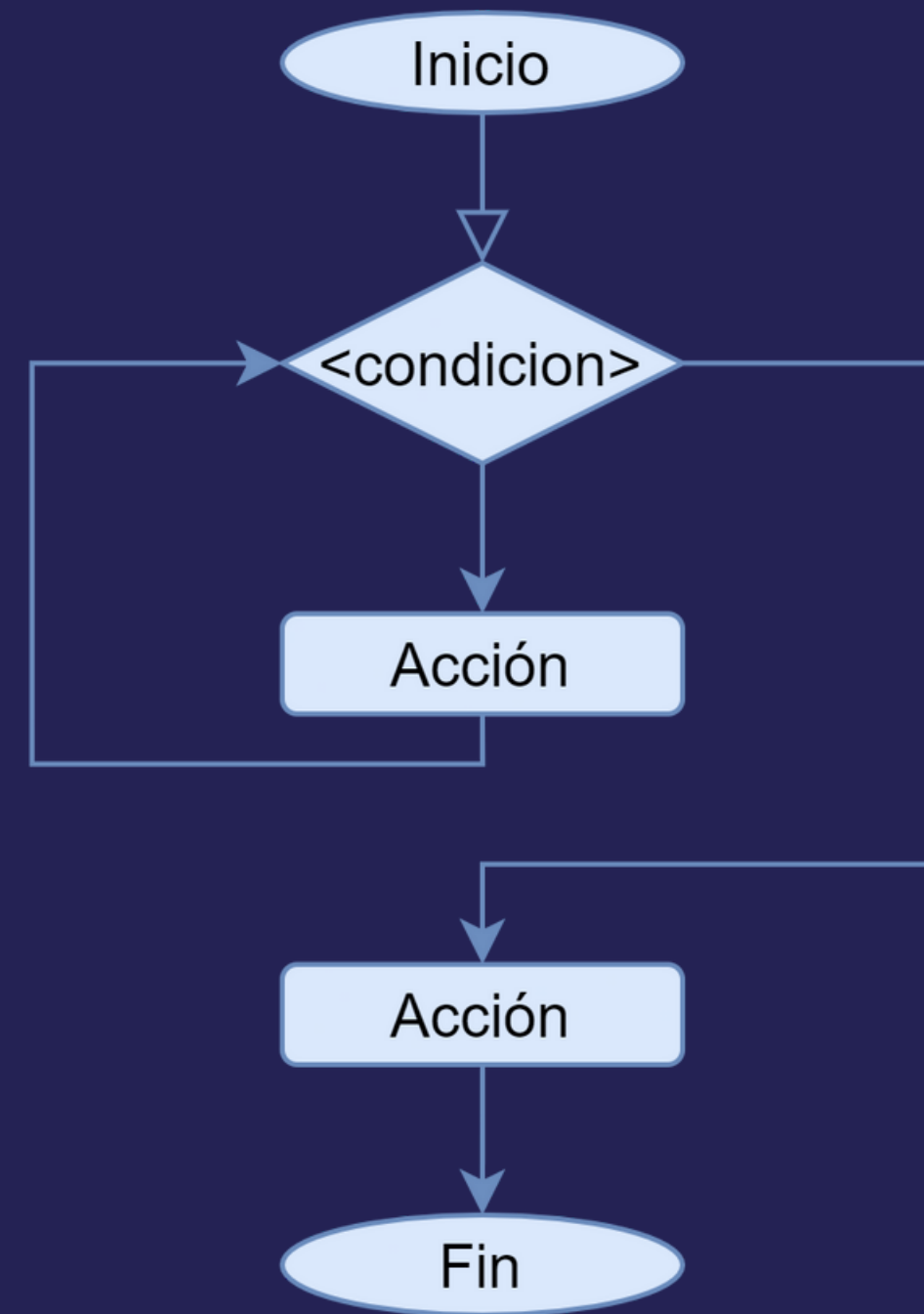
```
for <variable> in <elemento a recorrer>:  
    # do something
```



While



```
while <condition>:  
    # do something
```



Palabras reservadas útiles en bucles



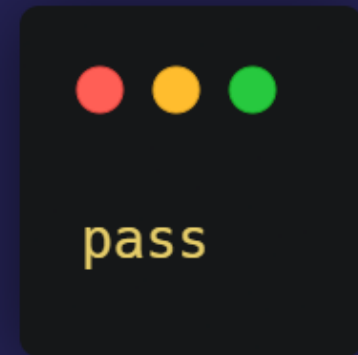
`break`

Te permite salir del bucle en cuanto se lea en el algoritmo



`continue`

Te permite saltar una iteración del bucle que se este leyendo



Te permite dejar una sección de código inconclusa en un segmento de código determinado

Listas

- Se puede expandir dinámicamente
- Pueden contener diferentes tipos de datos

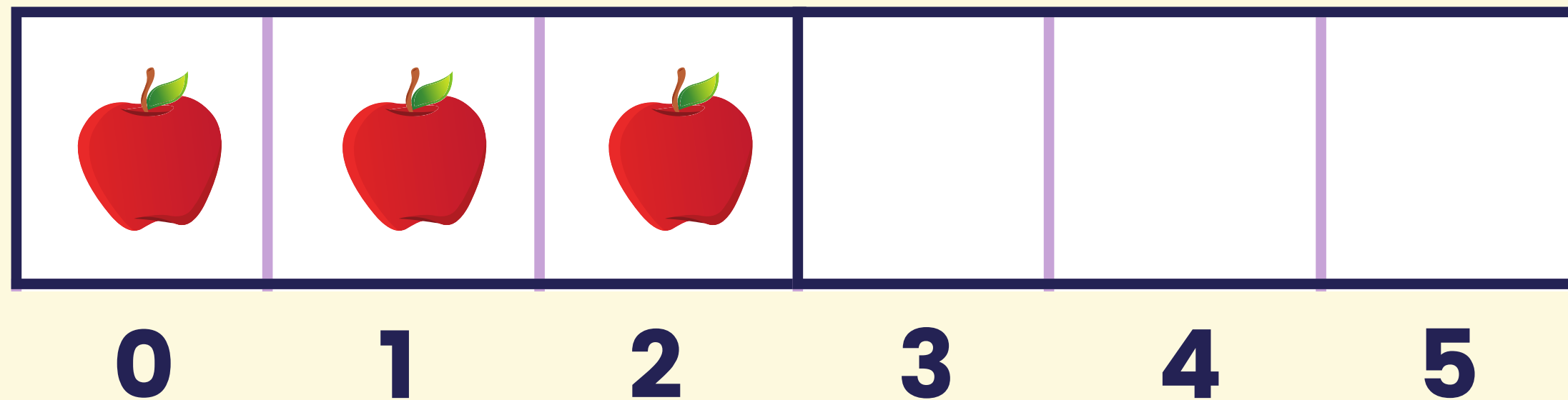
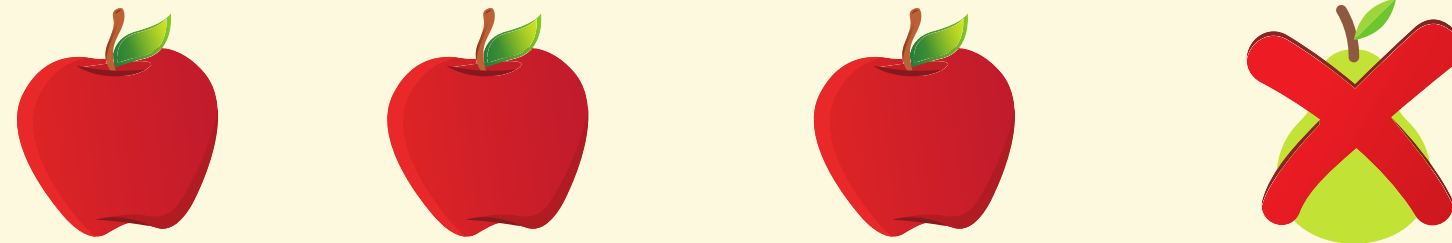
Crear una lista



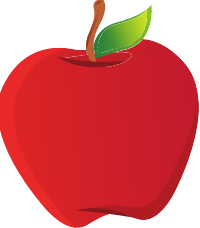
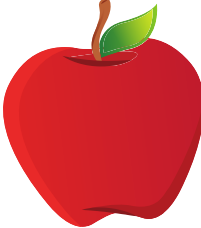
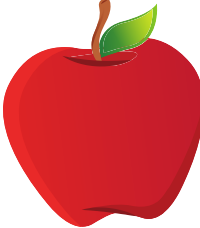
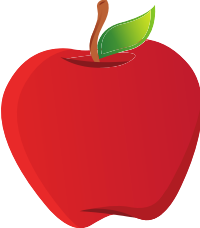
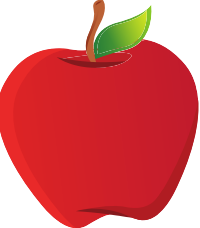
```
<nombre_de_la_lista> = [<elemento1>, <elemento2>, <elemento3>]
```

Arreglos



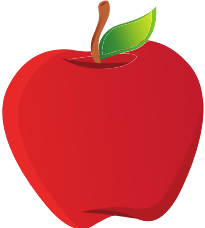

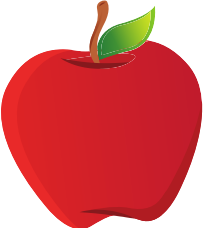
Un arreglo es una estructura de datos que almacena diferentes elemento con un mismo identificador. Solo pueden almacenar en su interior elementos del mismo tipo.



Memoria de computadora

| | | | | | |
|---|---|---|---|--|---|
| | | | | |  |
| |  | |  | | |
| | | | | | |
|  | |  | | | |

Memoria de computadora

| | | | | | |
|---|---|---|---|---|--|
| | | | | | |
| | | | | | |
| | | | | | |
|  |  |  |  |  | |


Trabajando con Listas

Acceder a una porción de la lista



```
miLista[0:3]
```

Agregar un elemento al final de la lista



```
miLista.append(<nuevo elemento>)
```

Extender una lista



```
miLista.extend([<elemento1>, <elemento2>, <elemento3>])
```

Obtener el índice de un elemento en la lista



```
miLista.index(<elemento_a_buscar>)
```

Remover un elemento de la lista



```
miLista.remove(<indice o nombre>)
```

Remover último miembro de la lista



```
miLista.pop()
```

Matrices

Es un arreglo de $M \times N$ elementos organizados en dos dimensiones donde "M" es el número de filas y "N" es el número de columnas

| | 0 | 1 | 2 |
|---|---|----|---|
| 0 | 5 | 12 | 3 |
| 1 | 4 | 6 | 9 |

$$(0, 1) = 12$$

$$(0, 2) = 3$$

$$(1, 0) = 4$$

$$(1, 1) = 6$$

Tuplas

- Son inmutables
- Pueden contener diferentes tipos de datos

Ventajas

- Más rápidas
- Menos espacio
- Pueden utilizarse como claves de diccionarios

Crear una tupla



```
<nombre de la lista> = (<elemento1>, <elemento2>, <elemento3>)
```

Convertir tupla a lista



```
list(<nombre de la tupla>)
```

Convertir lista a tupla



```
tuple(<nombre de la lista>)
```


Diccionarios

- Almacena valores de diferente tipo (enteros, strings, decimales, lista, diccionarios)
- Los datos se almacenan asociados a una clave de tal forma que se crea una asociación de tipo clave:valor
- Los elementos almacenados no están ordenados.

Crear un diccionario



```
miDiccionario = {<clave1>: <valor1>, <clave2>:<valor2>}
```

Añadir o sobrescribir un elemento



```
miDiccionario[<nueva clave o clave ya existente>] = <valor>
```

Eliminar un elemento



```
del midiccionario[<clave>]
```

Imprimir las claves



```
midiccionario.keys()
```

Imprimir valores



```
midiccionario.values()
```

Limpiar el diccionario completo



```
midiccionario.clear()
```

Obtener la longitud de un diccionario



```
len(midiccionario)
```

Set

- Colección de datos no ordenados

Crear un set



```
set = {1, 2, 3}
```