



EIN413B

Programación



UVA 6

Listas



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA



¿Dónde estamos?



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

EIN413B Programación / ELI109_A Introducción a la Programación

Sede Viña del Mar

Planificación Asignatura 2025-1

SEMANA	LUNES	MARTES	MIERCOLES	JUEVES	VIERNES	SABADO	DOMINGO
1	10-03-2025	11-03-2025	12-03-2025	13-03-2025	14-03-2025	15-03-2025	16-03-2025
2	17-03-2025	18-03-2025	19-03-2025	20-03-2025	21-03-2025	22-03-2025	23-03-2025
3	24-03-2025	25-03-2025	26-03-2025	27-03-2025	28-03-2025	29-03-2025	30-03-2025
4	31-03-2025	01-04-2025	02-04-2025	03-04-2025	04-04-2025	05-04-2025	06-04-2025
5	07-04-2025	08-04-2025	09-04-2025	10-04-2025	11-04-2025	12-04-2025	13-04-2025
6	14-04-2025	15-04-2025	16-04-2025	17-04-2025	18-04-2025	19-04-2025	20-04-2025
7	21-04-2025	22-04-2025	23-04-2025	24-04-2025	25-04-2025	26-04-2025	27-04-2025
8	28-04-2025	29-04-2025	30-04-2025	01-05-2025	02-05-2025	03-05-2025	04-05-2025
9	05-05-2025	06-05-2025	07-05-2025	08-05-2025	09-05-2025	10-05-2025	11-05-2025
10	12-05-2025	13-05-2025	14-05-2025	15-05-2025	16-05-2025	17-05-2025	18-05-2025
11	19-05-2025	20-05-2025	21-05-2025	22-05-2025	23-05-2025	24-05-2025	25-05-2025
12	26-05-2025	27-05-2025	28-05-2025	29-05-2025	30-05-2025	31-05-2025	01-06-2025
13	02-06-2025	03-06-2025	04-06-2025	05-06-2025	06-06-2025	07-06-2025	08-06-2025
14	09-06-2025	10-06-2025	11-06-2025	12-06-2025	13-06-2025	14-06-2025	15-06-2025
15	16-06-2025	17-06-2025	18-06-2025	19-06-2025	20-06-2025	21-06-2025	22-06-2025
16	23-06-2025	24-06-2025	25-06-2025	26-06-2025	27-06-2025	28-06-2025	29-06-2025
17	30-06-2025	01-07-2025	02-07-2025	03-07-2025	04-07-2025	05-07-2025	06-07-2025
18	07-07-2025	08-07-2025	09-07-2025	10-07-2025	11-07-2025	12-07-2025	13-07-2025

Semana	Tema	Control	SMOJ	Tarea
1	UVA 1: Python - Programas Secuenciales	-	-	-
2	UVA 1: Python - Programas Secuenciales	C1	UVA1	
3	UVA 2: Condicionales	C2	UVA2	TA1
4	UVA 3: Ciclos			
5	UVA 3: Ciclos	C3	UVA3	
6	UVA 4: Strings	C4	UVA4	
7	UVA 5: Funciones	C5	UVA5	TA2
8	Preparación Certamen 1			
9	Certamen 1 : Jueves 08/05 17:30 hrs.			
10	UVA 6: Listas	C6	UVA6	
11	Vacaciones Estudiantes			
12	UVA 6: Listas			TA3
13	UVA 7: Diccionarios	C7	UVA7	
14	UVA 8: Procesamiento de Texto y Archivos	C8	UVA8	TA4
15	Preparación Certamen 2			
16	Certamen 2 : Jueves 26/06 17:30 hrs.			
17	No hay más clases ni ayudantías			
18	Certamen Recuperativo : martes 08/07 17:30 hrs.			



Objetivos



1. Almacenar y procesar elementos en listas, recorriéndolas, agregando, buscando y borrando elementos.
2. Implementar algoritmos básicos sobre listas.
3. Resolver problemas que requieran almacenar y procesar elementos en listas alterando a conveniencia el orden de la estructura, o construyendo nuevas listas a partir de otras existentes.
4. Resolver problemas que requieran el tratamiento de listas de listas, construyendo y recorriendo las estructuras de datos a conveniencia (patrón de ciclo anidado).

¿Qué es una Lista?

Una lista en Python es una estructura de datos que sirve para guardar varios valores en una sola variable, manteniendo un orden y permitiendo modificar sus valores posteriormente.



UNIVERSIDAD TECNICA
FEDERICO SANTA MARIA

❖ Características:

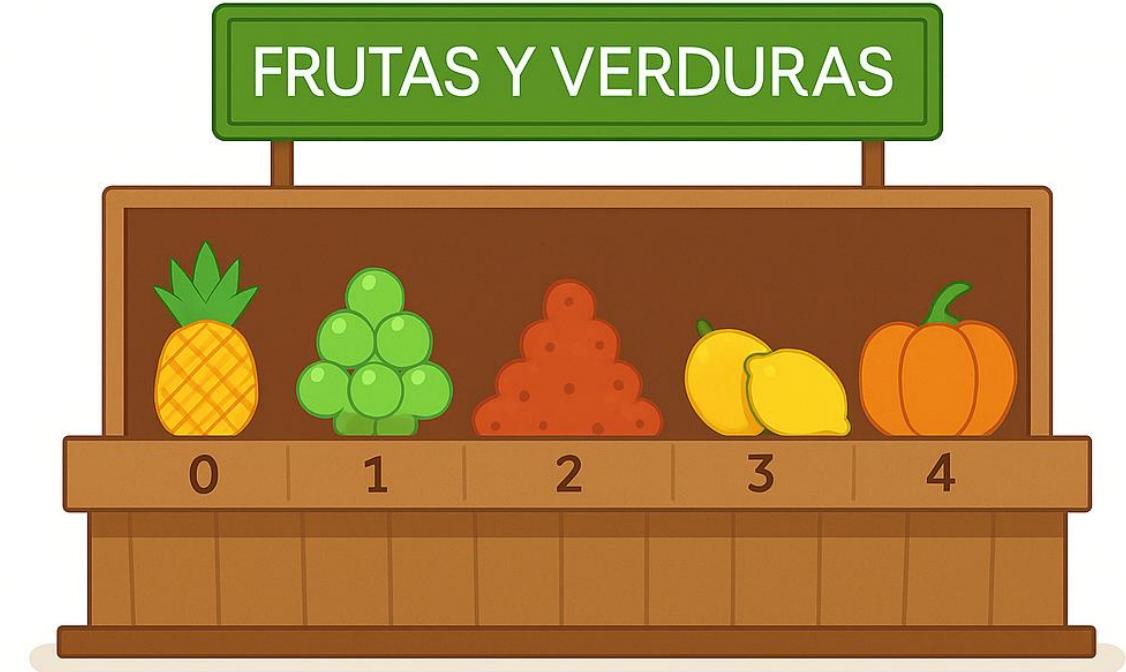
- **Ordenada:** Cada elemento tiene una posición (índice).
- **Mutable:** Puedes cambiar, agregar o eliminar elementos.
- **Flexible:** Puede mezclar distintos tipos de datos.
- **Accesible:** Puedes acceder a cada elemento por su índice.

¿Qué es una Lista?



💡 Características:

- **Ordenada:** Cada elemento tiene una posición (índice).
- **Mutable:** Puedes cambiar, agregar o eliminar elementos.
- **Flexible:** Puede mezclar distintos tipos de datos.
- **Accesible:** Puedes acceder a cada elemento por su índice.



¿Qué es una Lista?



```
# Lista de frutas y verduras
frutas_y_verduras = ['piña', 'uvas', 'tomates', 'limones', 'zapallo']

# Mostrar la lista completa
print(frutas_y_verduras)

# Mostrar cada elemento
for elemento in frutas_y_verduras:
    print(elemento)
```



Creación de listas

- □ ×

```
lista_vacia = []
lista_con_elementos = [10, 20, 30]
lista_desde_range = list(range(5)) # [0, 1, 2, 3, 4]
lista_desde_cadena = list("Python") # ['P', 'y', 't', 'h', 'o', 'n']
```

Acceso a elementos

- □ ×

```
print(frutas_y_verduras[1]) # Acceso al segundo elemento  
print(frutas_y_verduras[0:2]) # Acceso a una porción de la lista
```

- □ ×

```
# Lista de frutas y verduras  
frutas_y_verduras = ['piña', 'uvas', 'tomates', 'limones', 'zapallo']
```



Operadores con listas

```
1 lista1 = ["Rojo", "Azul"]
2 lista2 = ["Viernes", "Lunes"]
3 print(lista1 + lista2) # Concatenación
4 print(lista1 * 3) # Repetición
5 print("Amarillo" in lista1) # Verificación
6 del lista1[0] # Eliminación de elemento
7
```

```
['Rojo', 'Azul', 'Viernes', 'Lunes']
['Rojo', 'Azul', 'Rojo', 'Azul', 'Rojo', 'Azul']
False
```

Longitud de la lista: len(lista)

Suma de los elementos: sum(lista) (solo funciona con números)

Valor mínimo y máximo: min(lista), max(lista)

```
1 frutas_y_verduras = ['piña', 'uvas', 'tomates', 'limones', 'zapallo']
2 print(len(frutas_y_verduras))
3 print(min(frutas_y_verduras))
4 print(max(frutas_y_verduras))
```

```
5
limones
zapallo
```



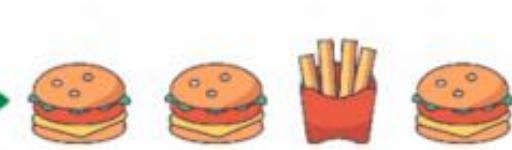
- Agregar un elemento al final: `lista.append(x)`
- Contar apariciones de un elemento: `lista.count(x)`
- Obtener el índice de un elemento: `lista.index(x)`
- Eliminar un elemento: `lista.remove(x)`
- Invertir la lista: `lista.reverse()`
- Ordenar la lista: `lista.sort()`
- Insertar un elemento en una posición específica: `lista.insert(pos, x)`

Métodos

```
1
2 frutas_y_verduras = ['piña', 'uvas', 'tomates', 'limones', 'zapallo']
3
4 frutas_y_verduras.append('manzana')
5 print(frutas_y_verduras)
6
7 print(frutas_y_verduras.count('uvas'))
8 print(frutas_y_verduras.index('tomates'))
9
10 frutas_y_verduras.remove('limones')
11 print(frutas_y_verduras)
12
13 frutas_y_verduras.reverse()
14 print(frutas_y_verduras)
15
16 frutas_y_verduras.sort()
17 print(frutas_y_verduras)
18
19 frutas_y_verduras.insert(2, 'pera')
20 print(frutas_y_verduras)
```

```
→ ['piña', 'uvas', 'tomates', 'limones', 'zapallo', 'manzana']
1
2
['piña', 'uvas', 'tomates', 'zapallo', 'manzana']
['manzana', 'zapallo', 'tomates', 'uvas', 'piña']
['manzana', 'piña', 'tomates', 'uvas', 'zapallo']
['manzana', 'piña', 'pera', 'tomates', 'uvas', 'zapallo']
```

Métodos

- **.append(burger)** → 
- **.count(burger)** → 2
- **.copy()** → 
- **.index(fries)** → 2
- **.reverse()** → 
- **.remove(fries)** → 
- **.insert(1, bubble tea)** → 

Con for: Iterar directamente sobre los elementos.



```
1 frutas_y_verduras = ['piña', 'uvas', 'tomates', 'limones', 'zapallo']
2
3 for elemento in frutas_y_verduras:
4     print(elemento)
```



piña
uvas
tomates
limones
zapallo

Recorridos de listas

Con while: Usar un índice para recorrer la lista.



```
1 frutas_y_verduras = ['piña', 'uvas', 'tomates', 'limones', 'zapallo']
2 i = 0
3 while i < len(frutas_y_verduras):
4     print(frutas_y_verduras[i])
5     i += 1
```



piña
uvas
tomates
limones
zapallo

- **Asignación directa:** `lista_copia = lista_original` crea una referencia.
- **Copia superficial:** `lista_copia = lista_original.copy()` crea una nueva lista con los mismos elementos.



```
1 frutas_y_verduras = ['piña', 'uvas', 'tomates', 'limones', 'zapallo']
2 verduleria = frutas_y_verduras.copy()
3 print(verduleria)
```

→ ['piña', 'uvas', 'tomates', 'limones', 'zapallo']

Uso de choice de la librería random



- Selecciona un elemento aleatorio de una lista.

```
1 import random
2 frutas_y_verduras = ['piña', 'uvas', 'tomates', 'limones', 'zapallo']
3 random.choice(frutas_y_verduras)
4
```

'tomates'

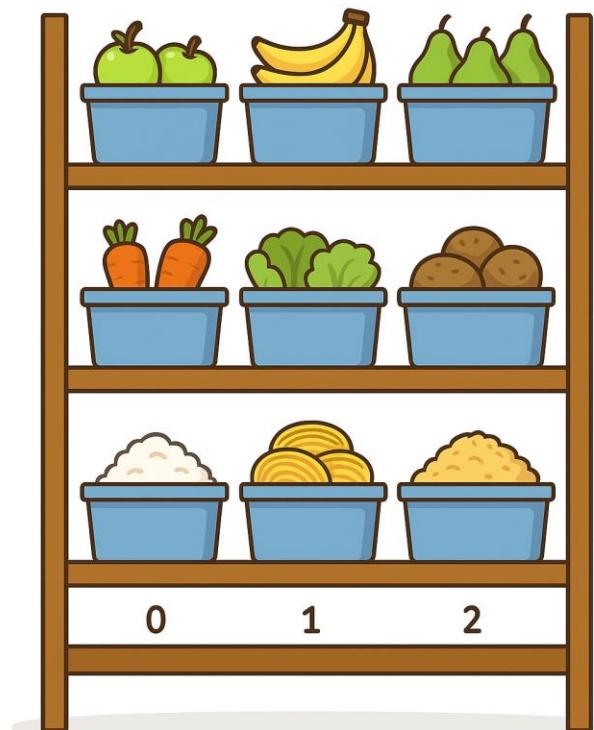
Listas de listas (matrices)

Una lista de listas en Python es una estructura que permite guardar listas dentro de otra lista. Es decir, es una lista donde cada elemento también es una lista.

Imagina un **estante de bodega** con varias **repisas**.

Cada repisa tiene **cajas**, y dentro de cada caja hay alimentos.

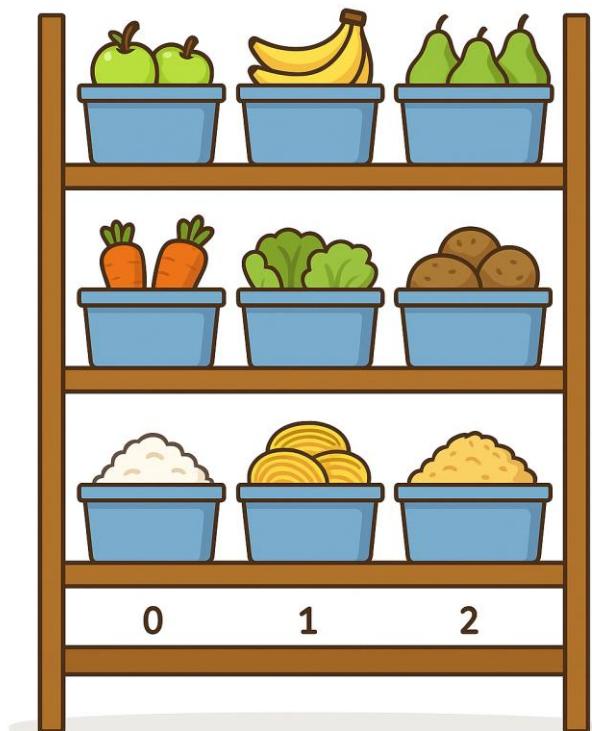
- El **estante completo** es la **lista principal**.
- Cada **repisa** es una **sublista** (otra lista dentro de la principal).
- Cada **objeto dentro de una caja** es un **elemento** de esa sublista.

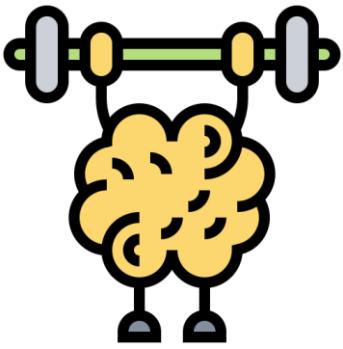


Listas de listas (matrices)

```
1 alimentos = [
2     ["manzana", "banana", "pera"],           # Frutas (índice 0)
3     ["zanahoria", "lechuga", "papa"],        # Verduras (índice 1)
4     ["arroz", "fideos", "avena"]             # Cereales (índice 2)
5 ]
6
7 print(alimentos[0]) #devuelve la lista de frutas.
8
9 print(alimentos[1][2])#devuelve "papa" (3a verdura).
```

```
['manzana', 'banana', 'pera']
papa
```





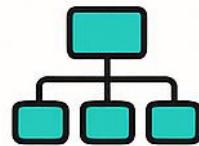
Ejercicio



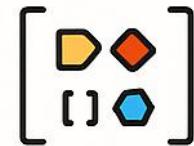
1. Crea una lista llamada calificaciones que contenga 10 números aleatorios entre 1 y 10. Pista: Usa random.randint().
2. Recorre la lista y calcula la calificación promedio.
3. Muestra la calificación más alta y la más baja de la lista.



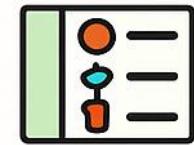
¿Qué aprendimos?



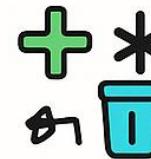
Qué es una estructura de datos



Qué es una lista en Python



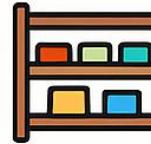
Cómo crear listas y acceder a sus elementos



Cómo modificar y operar listas



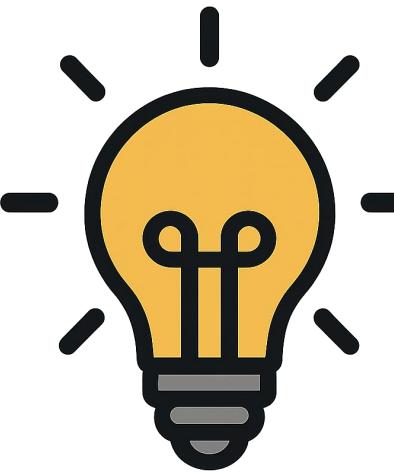
Cómo recorrer listas



Cómo usar listas de listas



Aplicaciones prácticas



¿Qué aprendimos?



1. **Qué es una lista en Python:** Aprendimos que una lista es una colección ordenada, modificable y que puede contener cualquier tipo de dato.
2. **Cómo crear listas y acceder a sus elementos:** Utilizamos índices, rebanado ([a:b]) y funciones como len() o in.
3. **Cómo modificar y operar listas:** Usamos operadores (+, *, del) y métodos como .append(), .remove(), .sort().
4. **Cómo recorrer listas:** Aplicamos ciclos for y while para iterar sobre sus elementos.
5. **Cómo usar listas de listas:** Representamos estructuras más complejas como estantes con niveles, tablas o matrices.