



ESCUELA DE INGENIERÍA



# Comparación de estructuras secuenciales



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DE CHILE



## **Cristian Ruz Ruz**

Profesor Asistente Adjunto, Departamento de  
Ciencia de la Computación  
Facultad de Ingeniería  
Pontificia Universidad Católica de Chile

# Índice




Comparación de estructuras secuenciales



Medición de eficiencia de listas versus colas

# Introducción

- Características de estructuras secuenciales. 
- Ventajas y desventajas entre estructuras secuenciales.
- Estructura secuencial más apropiada para un problema.

# **Comparación de estructuras secuenciales**

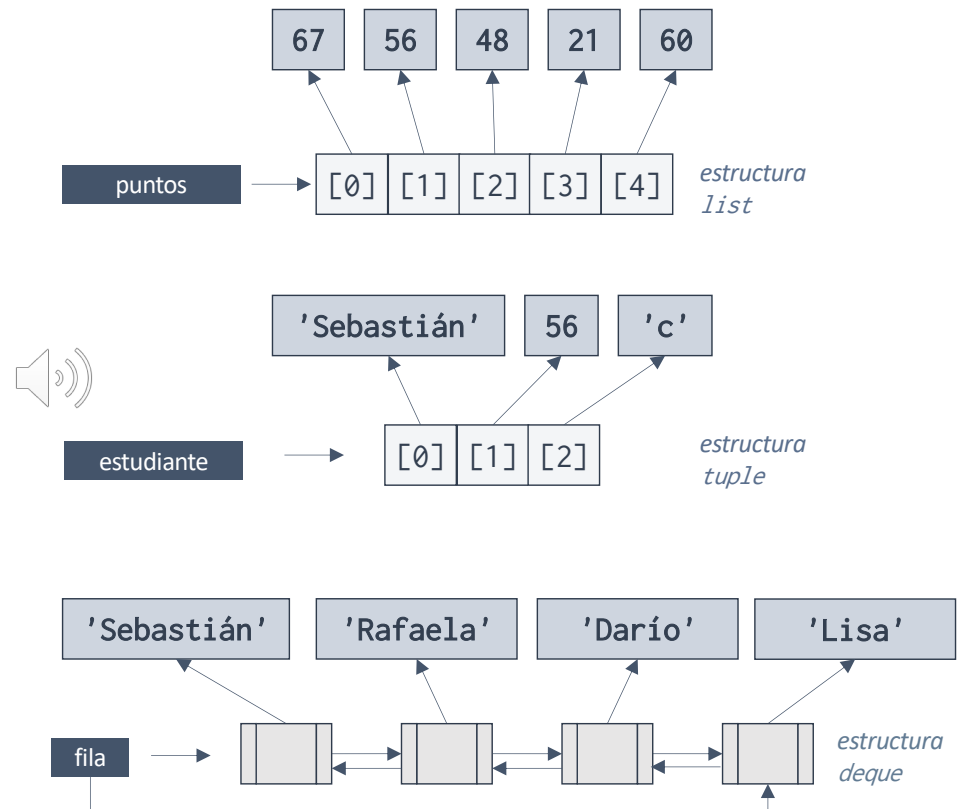


# Estructuras secuenciales

Estructuras ordenadas de datos


Almacenan cualquier tipo de elementos

Operaciones para consultas y manipulación de elementos



## Listas vs tuplas vs colas

list
<code>lista = [67, 56, 48, 21]</code>
Mutable
Indexable
Permite <i>slicing</i>
Agregar/eliminar al inicio: ✓
Agregar/eliminar al final: ✓✓✓
Insertar/eliminar en posición i: ✓
Leer cualquier posición: ✓✓✓
Útil para elementos homogéneos

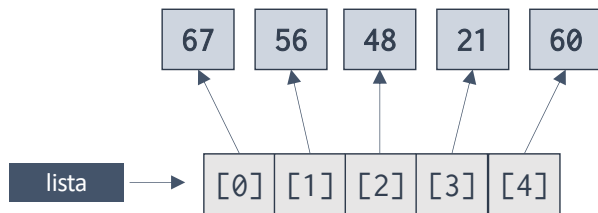
tuple
<code>tupla = (67, 56, 48, 21)</code>
Inmutable
Indexable 
Permite <i>slicing</i>
Agregar/eliminar al inicio: ✗
Agregar/eliminar al final: ✗
Insertar/eliminar en posición i: ✗
Leer cualquier posición: ✓✓✓
Útil para elementos heterogéneos relacionados

deque
<code>cola = deque([67, 56, 48, 21])</code>
Mutable
Indexable
No permite <i>slicing</i>
Agregar/eliminar al inicio: ✓✓✓
Agregar/eliminar al final: ✓✓✓
Insertar/eliminar en posición i: ✓
Leer cualquier posición: ✓
Útil para elementos homogéneos con modificaciones en los extremos

# Listas vs tuplas vs colas

list

```
lista = [67, 56, 48, 21, 60]
```



Agregar/eliminar al inicio: ✓

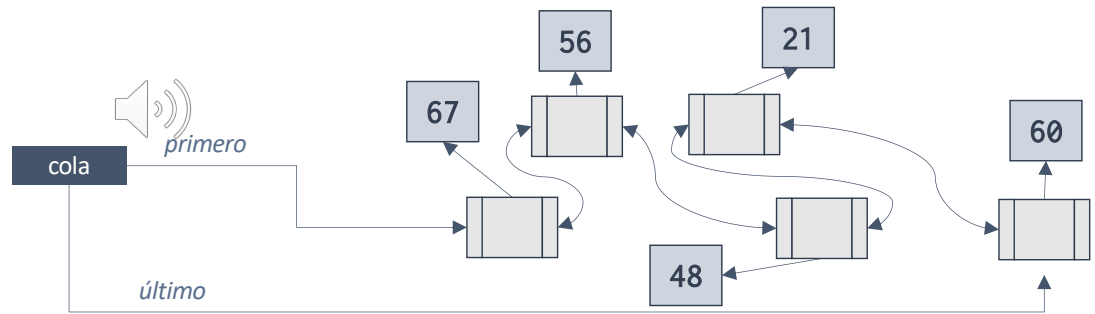
Agregar/eliminar al final: ✓✓✓

Insertar/eliminar en posición i: ✓

Leer cualquier posición: ✓✓✓

list

```
cola = deque([67, 56, 48, 21, 60])
```



Agregar/eliminar al inicio: ✓✓✓

Agregar/eliminar al final: ✓✓✓

Insertar/eliminar en posición i: ✓

Leer cualquier posición: ✓



# **Eficiencia de listas versus colas**



## Midiendo tiempo de acceso

1

### Acceder a la posición del medio en deque y en list

```
from collections import deque
from time import time
number_deque = deque(range(1000000))
number_list = list(range(1000000))

start_time = time()
number_deque[500000]
finish_time = time()
deque_time = finish_time - start_time
print(f"Buscar el elemento 500000 en el deque demoró {deque_time:.6f} seg")

start_time = time()
number_list[500000]
finish_time = time()
list_time = finish_time - start_time
print(f"Buscar el elemento 500000 en la lista demoró {list_time:.6f} seg")
print(f"La búsqueda en deque fue {deque_time/list_time:.2f} veces el tiempo de list.")
```

Buscar el elemento 500000 en el deque demoró 0.004431 seg  
Buscar el elemento 500000 en la lista demoró 0.000060 seg  
La búsqueda en deque fue 74.35 veces el tiempo de list

## Midiendo tiempo de acceso

2

### Sacar los primeros 1000 elementos del inicio

```
number_deque = deque(range(10000000))
number_list = list(range(10000000))

start_time = time()
for i in range(1000):
    number_deque.popleft()
finish_time = time()
deque_time = finish_time - start_time
print(f"Sacar los primeros 1000 de deque demoró {deque_time:.6f} seg")

start_time = time()
for i in range(1000):
    number_list.pop(0)
finish_time = time()
list_time = finish_time - start_time
print(f" Sacar los primeros 1000 de lista demoró {list_time:.6f} seg")
print(f"La extracción en list fue {deque_time/list_time:.2f} veces el tiempo de deque.")
```



Sacar los primeros 1000 de deque se demoró 0.000152 seg  
Sacar los primeros 1000 de lista se demoró 7.044542 seg  
La extracción en list fue 46457.47 veces el tiempo de deque.

## Síntesis

- Características de estructuras secuenciales.
- Cómo se explica la eficiencia de cada estructura.
- Comparación de tiempo entre listas y colas.
- Todo depende del uso que le vamos a dar nuestra estructura.



## Referencias bibliográficas

- Data Structures. The Python Tutorial, v3.8.2rc1. Recuperado de:  
<https://docs.python.org/3/tutorial/datastructures.html>
- Container datatypes. The Python Standard Library, v3.8.2rc1. Recuperado de:  
<https://docs.python.org/3/library/collections.html>



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DE CHILE