



ESCUELA DE INGENIERÍA



# Colas y operaciones



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DE CHILE



## Cristian Ruz Ruz

Profesor Asistente Adjunto, Departamento de  
Cie<sup>nc</sup>ia de la Computación  
Facultad de Ingeniería  
Pontificia Universidad Católica de Chile

# Índice



Concepto de cola



Estructura deque



Operaciones sobre colas

# Introducción

- Conocer la estructura de datos cola.
- Crear y manipular colas en Python.
- Qué operaciones se pueden aplicar sobre colas.





# Concepto de cola

# ¿Qué es una cola?

Colección de datos

Estructura de datos

Estructura **ordenada**

Acceso eficiente **a sus extremos**



## Colas en Python: deque()

1

### Creación de una lista vacía

```
from collections import deque  
cola = deque()
```

2

### Mostrar el contenido

```
print(cola)  
deque([])
```

3

### Tipo de dato

```
print(type(cola))  
<class 'collections.deque'>
```

## 4

### Creación de una cola con elementos

```
profesores = deque(['Jaime', 'Cristian', 'Denis'])
cola_vacia = deque([])
mezcla = deque(['Chile', 8, 4.15, 'Abril'])
sublista = deque([15, ['Arica', 9], 'Febrero'])
```

## Colas en Python: deque()



```
print(profesores)
print(cola_vacia)
print(mezcla)
print(sublista)
```

```
deque(['Jaime', 'Cristian', 'Denis'])
deque([])
deque(['Chile', 8, 4.15, 'Abril'])
deque([15, ['Arica', 9], 'Febrero'])
```

## Colas en Python: deque()

5

### Creación por concatenación

```
otoño = deque(['Abril', 'Mayo', 'Junio'])
invierno = deque(['Julio', 'Agosto'])
frío = otoño + invierno
más_frión = invierno * 2
```

```
print(frío)
print(más_frión)
```

```
deque(['Abril', 'Mayo', 'Junio', 'Julio', 'Agosto'])
deque(['Julio', 'Agosto', 'Julio', 'Agosto'])
```

## Consultas sobre una cola

índices →

0 1 2 3 4 5 6 7

```
mezcla = deque(['Chile', 8, 4.15, 'Abril', 3, 5, '2020', 'MOOC'])
```

### Longitud de una lista

```
largo = len(mezcla)  
print(largo)
```

8

### Elemento en posición *i*



```
print(mezcla[0])  
print(mezcla[4])  
print(mezcla[7])  
print(mezcla[-2])
```

Chile  
3  
MOOC  
2020

### Extrayendo porciones (*slicing*)



```
print(mezcla[2:6])  
print(mezcla[:3])  
print(mezcla[5:])  
print(mezcla[1:6:2])
```

*deques* no soportan *slicing*

## Modificar elementos de una cola

`deque[i] = ...`

1



Modificar elementos en la posición *i*

```
mezcla[3] = 'Mayo'  
mezcla[4] += 5  
print(mezcla)
```

```
deque(['Chile', 8, 4.15, 'Mayo', 8, 5, '2020', 'MOOC'])
```

```
mezcla = deque(['Chile', 8, 4.15, 'Abril', 3, 5, \  
'2020', 'MOOC'])
```

# Recorriendo colas

for \_ in deque:

```
mezcla = deque(['Chile', 8, 4.15, 'Abril', 3, 5, \  
'2020', 'MOOC'])
```

1

Recorriendo con for \_ in range:

```
for i in range(len(mezcla)):  
    print(mezcla[i])
```

- Recorrido usando un rango de índices.
- Útil para  buscar elementos y recordar su posición

2

Recorriendo con for \_ in lista:

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

- Recorrido por elemento.
- Fácil para observar cada elemento si no importa la posición

# Operaciones sobre colas: agregar y sacar desde el final

1

## Agregar al final: append

```
opciones = deque(['Sí', 'No'])
opciones.append('No sé')
print(opciones)
```

```
deque(['Sí', 'No', 'No sé'])
```

2

## Sacar desde el final: pop



```
cuando = deque(['Mañana', 'Pasado', 'Hoy'])
eliminado = cuando.pop()
print(cuando)
print(eliminado)
```

```
deque(['Mañana', 'Pasado'])
Hoy
```

# Operaciones sobre colas: agregar y sacar desde el inicio

1

## Agregar al inicio: appendleft

```
opciones = deque(['Sí', 'No'])
opciones.appendleft('No sé')
print(opciones)
```

```
deque(['No sé', 'Sí', 'No'])
```

2

## Sacar desde el inicio: popleft



```
cuando = deque(['Mañana', 'Pasado', 'Hoy'])
eliminado = cuando.popleft()
print(cuando)
print(eliminado)
```

```
deque(['Pasado', 'Hoy'])
Mañana
```

## Operaciones sobre colas: insertar en cualquier posición

1

### Insertar en posición $i$ : insert

```
opciones = deque(['Si', 'No', 'No sé'])
opciones.insert(2, 'Quizás')
print(opciones)
```

```
deque(['Si', 'No', 'Quizás', 'No sé'])
```

# Operaciones sobre colas: encontrar y eliminar un elemento

## Encontrar posición de elemento: index

```
opciones = deque(['Si', 'No', 'Quizás', 'No sé'])  
print(opciones.index('Quizás'))  
print(opciones)
```

2

## Eliminar elemento: remove



```
cuento = deque(['Mañana', 'Pasado', 'Hoy'])  
cuento.remove('Pasado')  
print(cuento)
```

```
deque(['Mañana', 'Hoy'])
```

## Síntesis

- Concepto de cola.
- Implementación en Python: deque.
- Consultas: longitud, índices.
- Recorrido: for ... in.
- Operaciones: append, pop, appendleft, popleft, index, insert, remove.



## **Referencias bibliográficas**

- Container datatypes. The Python Standard Library, v3.8.2rc1. <https://docs.python.org/3/library/collections.html>
  
- Imagen diapositiva 8: “Retráctil Del Cinturón De Puntal Multitud Cola por Juhele Pixabay Licencia. Recuperado de: <https://pixabay.com/es/vectors/retr%C3%A1ctil-del-cintur%C3%B3n-de-puntal-4326819/>. Pixabay License.
  
- Imagen diapositiva 8: “Queues at London Luton Airport” por oatsy40 licencia bajo CC BY-2.0. Recuperado de: [https://commons.wikimedia.org/wiki/File:Queue\\_\(42119147194\).jpg](https://commons.wikimedia.org/wiki/File:Queue_(42119147194).jpg).



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA  
DE CHILE