



Colas

Una cola es una estructura de datos lineal y sigue un patrón llamado “FIFO” (First IN, First OUT). Osea el primero que entra es el primero que sale, Para hacer una analogía con el mundo real tenemos que pensar en la cola de espera de un banco.



FIG 00: representación de FIFO

FIFO es igual a una fila de personas esperando su turno, por más clientes que lleguen a esperar a la cola siempre se va a atender al primero que llegó.

Por qué existen las colas

- Sistemas donde el orden de procesamiento es crucial.
- Sistemas que requieren manejar a medida que van llegando.
- Sistemas donde entran datos con diferentes velocidades (VIDEO)



Advertencia: Se espera que el lector entienda que al igual que el mundo real cuando una persona es atendida en la cola desaparece (Se va de la cola).

Una cola no es más que una colección de Nodos, los cuales contienen datos y un enlace a otro nodo. Cada nodo contiene 2 elementos:

- Dato.
- Nodo siguiente.

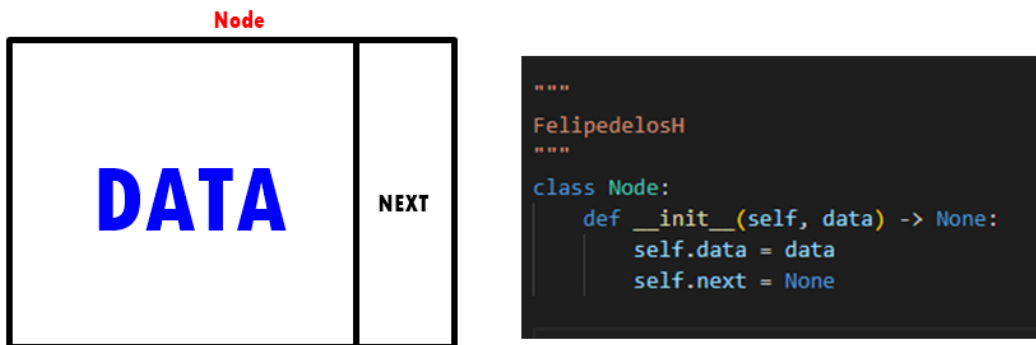


FIG 00: representación de un nodo al lado de su respectivo código.

Para poder utilizar esto necesitamos crear algo que se llama “Cola”

Una cola es una colección de nodos que nos va a permitir guardar la información, dicha clase contiene un apuntador y varios datos... si observa la siguiente figura usted podrá notar cómo funciona:

Cola

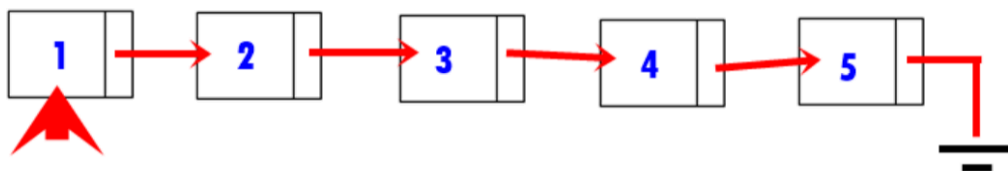


FIG 01: Ejemplo visual de una cola.

Las colas necesitan los siguientes atributos y métodos para poder funcionar:

- Se necesita un pivote el cual es el encargado de moverse para leer/escribir la información.
- Se necesita un método para encolar elementos.
- Se necesita un método para desencolar elementos.

- Se necesita un contador para obtener el total de elementos de la cola.

Que es el pivote

El pivote no es más que una instancia del nodo, pero no podemos empezar la cola instanciando un nodo sin una data (o al menos no es recomendable) el pivote es la cabeza “un nodo” y se encarga de moverse para recorrer la cola y poder agregar elementos.

El pivote/cabezal es el encargado de ser el punto de acceso a los datos de la cola sin él no existiría una forma de recorrerla.

```
class Queue:
    def __init__(self) -> None:
        self.pivot = None
        self.size = 0
```

Advertencia: el pivote inicia en None dado a que cuando la cola es creada no posee datos.

Como se encolan los datos

Yo he decidido hacerlo de manera iterativa. para agregar a la cola se tiene que partir de 2 casos base:

- Caso 1: la cola está vacía: La solución es simple se declara un nuevo nodo con la información a almacenar y luego el pivote se asigna a el nuevo nodo:

```
def addData(self, data):
    new_node = Node(data)
    if self.pivot == None:
        self.pivot = new_node
```

Advertencia: al final de este método se aumenta el contador del tamaño de la cola.

- Caso 2: La cola no está vacía en ese caso lo que hacemos es crear un nuevo nodo donde va a estar contenida el nuevo dato y con un while empezamos a recorrer los nodos de siguiente en siguiente hasta encontrar el que se encuentre vacío, luego lo guardamos en la siguiente posición vacía.

```
_copy = self.pivot

while _copy.next != None:
    _copy = _copy.next

_copy.next = new_node
```

Advertencia: se saca copia del pivote por precaución, para evitar borrar el original

Como desencolar los elementos

Como la lógica de las colas es que el primer elemento es el atendido y luego este desaparece. Lo que tenemos que hacer es guardar el dato en una variable temporal, luego pasar el pivote hacia el siguiente (Advertencia: como un elemento ha desaparecido hay que decrementar la variable que controla el tamaño de la lista).

```
data = self.pivot.data
self.pivot = self.pivot.next
self.size = self.size - 1

return data
```