

# Fundamentos de Queue en C#

## 1. Definición

En C#, `Queue<T>` es una colección genérica que implementa una estructura FIFO (First In, First Out).

- El primer elemento en entrar es el primero en salir.
- Ideal para procesar datos en el orden en que fueron recibidos (ej.: colas de impresión, atención a clientes).

Namespace:

```
using System.Collections.Generic;
```

## 2. Declaración y creación

```
Queue<string> printQueue = new Queue<string>();
```

- `<string>` indica que la cola almacenará cadenas.
- Puede almacenar tipos primitivos u objetos personalizados.

## 3. Operaciones principales de Queue<T>

| Método                     | Descripción                                       |
|----------------------------|---|
| <code>Enqueue(item)</code> | Agrega un elemento al final de la cola.           |
| <code>Dequeue()</code>     | Elimina y devuelve el primer elemento de la cola. |
| <code>Peek()</code>        | Devuelve el primer elemento sin eliminarlo.       |
| <code>Count</code>         | Obtiene la cantidad de elementos en la cola.      |
| <code>Clear()</code>       | Elimina todos los elementos.                      |

## 4. Ejemplo de uso: Gestor de impresión

```
printQueue.Enqueue("Factura.pdf");  
Thread.Sleep(1000);
```

```
printQueue.Enqueue("Presentacion Marketing.pptx");  
Thread.Sleep(1000);
```

```
printQueue.Enqueue("Estadisticas.xlsx");  
Thread.Sleep(1000);
```

```
printQueue.Enqueue("Carta.docx");
```

- `Enqueue()` añade documentos a la cola.
- `Thread.Sleep(ms)` simula tiempo de espera.

## 5. Consulta de elementos sin eliminarlos

```
if (printQueue.Count > 0)  
{  
    string nextDocument = printQueue.Peek();  
    Console.WriteLine($"Siguiente documento: {nextDocument}");  
}
```

```
}
```

## 6. Procesamiento de la cola

```
while (printQueue.Count > 0)
{
    string currentDocument = printQueue.Dequeue();
    Console.WriteLine($"Imprimiendo: {currentDocument}");
    Thread.Sleep(1000);
}
```

- Dequeue() extrae en orden FIFO.
- Complejidad:  $O(1)$  para Enqueue y Dequeue.

## 7. Salida esperada (ejemplo)

--- Gestor de Impresión ---

Enviando documentos a la impresora:

- 'Factura.pdf' enviada
- 'Presentacion Marketing.pptx' enviada
- 'Estadisticas.xlsx' enviada
- 'Carta.docx' enviada

Documentos en cola: 4

El siguiente documento a imprimir es: 'Factura.pdf'

Iniciando impresión de documentos:

- Imprimiendo: 'Factura.pdf'
- Imprimiendo: 'Presentacion Marketing.pptx'
- Imprimiendo: 'Estadisticas.xlsx'
- Imprimiendo: 'Carta.docx'

Todos los documentos han sido impresos.

Documentos restantes en cola: 0

## Buenas prácticas

- Usar Queue<T> cuando se necesite mantener el orden de llegada.
- Verificar Count > 0 antes de llamar a Dequeue() o Peek() para evitar excepciones.
- Si se necesita acceso aleatorio a los elementos, considerar List<T> en su lugar.