

# PROGRAMACIÓN

## Guía de estudio 13:

# Hay mucha cola en el super

## Colas en C#


<b>Nivel:</b> Secundario - Modalidad Educación Técnico-Profesional.
<b>Ciclo:</b> Segundo ciclo.
<b>Especialidades:</b> Informática, Programación.

### Introducción

Así como vimos pilas en la guía anterior, en esta guía toca ver "colas". Las colas, a diferencia de las pilas, están asociadas con la estructura FIFO (First In First Out). Se comportan exactamente como las colas del supermercado: el primero que llega es el primero que será atendido. Veamos cómo se usan y se crean.

**¿Qué estamos aprendiendo?:** Cómo crear colas y utilizarlas en c#. Funciones relacionadas para utilizar colas.

### Recursos:

Video nro 26 del curso:	
<a href="https://youtu.be/ab8jklLvt-Q">https://youtu.be/ab8jklLvt-Q</a>	

Video nro 27 del curso: <a href="https://youtu.be/cb4twl4zSTM">https://youtu.be/cb4twl4zSTM</a>	
--	---

## Secuencia didáctica

Mirá los siguientes vídeos en los cuales se explica el tema que ya hemos introducido: colas.

<a href="https://youtu.be/ab8jklLvt-Q">https://youtu.be/ab8jklLvt-Q</a>	
<a href="https://youtu.be/cb4twl4zSTM">https://youtu.be/cb4twl4zSTM</a>	

## Actividades

Después de mirar el video del link o del QR, te invitamos a que reflexiones con las siguientes preguntas.

*No olvides escribir tus respuestas, te facilitarán realizar trabajos prácticos y repasar los temas que hayas estudiado con esta guía.*

- 1) ¿Cuál es la diferencia entre peek y pop? ¿En donde creés haber visto este sistema aplicado en la vida real?
- 2) Hace un programa de atención en un centro médico. El programa deberá tener un menú de 3 opciones. La primera opción es para registrar un paciente nuevo y se deberá guardar el nombre del paciente. La segunda opción es para avisar que un consultorio fue liberado y se deberá mostrar el paciente que debería entrar al consultorio. La tercera opción es para salir del programa.
- 3) Hagamos un gestor de tareas (para organizarnos mejor). Cada vez que se ingrese un texto que sea diferente de "-1", se deberá guardar ese texto en una cola llamada

"tareas pendientes". Si el texto es "-1", se deberá mostrar la tarea más antigua que tenga pendiente, eliminándola de la lista. Si se ingresa "0" el programa finaliza.

- 4) Hacé un programa para ayudar a el restaurant "Don Manolo" a organizar la atención a los que vienen a comprar por mostrador.
- Si se ingresa la palabra "agregar", se deberá preguntar el nombre de la persona y almacenarlo para llamarlo cuando sea su turno.
  - Si se ingresa la palabra "llamar", se saca a la persona de la cola. Se le pregunta el pedido, y se guarda nombre y pedido (de la manera que mejor se te ocurra).
  - Si se ingresa "listo", se deberán preguntar el nombre, y tendrá que mostrar en pantalla el pedido que se está entregando, sacándolo de los pedidos pendientes.
  - Si se ingresa "salir" se deberá cerrar el programa.

## Para seguir aprendiendo:

Vamos a aprovechar este sector para mejorar un poco como programadores.

### Consejo 1:

Comentar TODO. Cuando estamos haciendo programas largos, y con varias funciones, todo lo que tenemos escrito puede ser bastante caótico. Es por eso que está bueno agregar comentarios indicando brevemente lo que se está haciendo (especialmente si usamos muchas variables o hacemos muchas operaciones seguidas) Recordar que para comentar en un renglón, usamos //. Si queremos poner un comentario de varios renglones podemos usar /\* para comenzar y \*/ para finalizar el comentario. Los comentarios también se usan para "anular" un pedazo de código sin borrarlo (al comentarlo, se deja de ejecutar ;pero sigue allí!)

### Consejo 2:

En c# (como en muchos otros lenguajes de programación) podemos abreviar nuestro código de múltiples maneras. Esto nos permite (a nosotros y a los que lean nuestro código) entender más fácil lo que está pasando. Los recortes que vamos a mostrar son válidos para if, for, y while.

Veamos el siguiente ejemplo:

```
original
if(algo == 0){
    Console.WriteLine("error");
}
```

*recorte 1:* `if(algo == 0) Console.WriteLine("error");`

Pueden ver que si solo ejecutamos 1 instrucción dentro del if, podemos sacar las llaves que lo rodean y dejarlo todo en la misma línea.

*recorte 2:* `if(!algo) Console.WriteLine("error");`

Si "algo" es una variable del tipo Bool, podemos evaluar si una variable es falsa o "0" simplemente poniendo el signo de exclamación antes de la variable.

Si queremos ver si es verdadera o 1, podemos hacer `if(algo) instrucción`, es decir, alcanza con el mismo nombre de la variable.