# http://vectorblog.org/wp-content/uploads/2013/06/Telehealth-cartoon-e1370345782337.jpgHoja de trabajo No. 2: Tipo de dato abstracto *Cola*

Objetivos de la hoja de trabajo

* Utilizar una estructura de datos donde el primer elemento ingresado será el primer elemento extraído: Cola.
* Introducir al estudiante al tipo de dato abstracto *Cola*.

Recomendación: trabajar en grupos de dos integrantes.

## Parte 1: Utilización de una *Cola*

Lectura:

* Secciones: *Queues, Definitions, Queue Operations*, del libro *Data Structures and Program Design in C++*, páginas 78 – 81.
* Referencia técnica de la clase Queue<T> en Microsoft Developer Network. Disponible en:  
  <https://msdn.microsoft.com/en-us/library/7977ey2c(v=vs.110).aspx>

1. Aparee las siguientes operaciones mencionadas en la lectura del libro con los métodos de la clase Queue:

|  |  |
| --- | --- |
| **Operaciones de Queue (tipo de dato abstracto definido en el libro)** | **Métodos de la clase Queue<T>** |
| Retrieve | Peek |
| Serve | Dequeue |
| Append | Queue |

1. ¿Cuál es la salida del siguiente código? Sugerencia: coloque dicho código en un programa en C#.  
   Queue<int> unaCola = new Queue<int>();  
   unaCola.Enqueue(345);  
   unaCola.Enqueue(29);  
   int unNumero = unaCola.Dequeque();  
   MessageBox.Show(unNumero.ToString());  
   int otroNumero = unaCola.Dequeque();  
   MessageBox.Show(otroNumero.ToString());

* el primer MessageBox.Show() muestra 345
* el segundo muestra 29

1. ¿Cuál es la salida del siguiente código? Sugerencia: coloque dicho código en un programa en C#.  
   Queue<int> otraCola = new Queue<int>();  
   try   
   {  
    otraCola.Enqueue(9);  
    int unNumero = unaCola.Dequeque();  
    MessageBox.Show(unNumero.ToString());  
    int otroNumero = unaCola.Dequeque();  
    MessageBox.Show(otroNumero.ToString());  
   }  
   catch(Exception unaExcepcion)  
   {  
    MessageBox.Show(“Ocurrió un error: ” + unaExcepcion.Message);  
   }

* lanza una InvalidOperationException

1. ¿Cómo se relacionan los elementos de una cola?

* Tiene secuencia igual que en una pila.

1. Indique las entradas, salidas y poscondición del método Enqueue de la clase Queue<T>.

* **Entradas:** dato de tipo T
* **Salidas:** nada
* **Poscondición:** añade el dato en la ultima posición de la cola.

1. Indique las entradas, salidas y poscondición del método Dequeue de la clase Queue<T>.

* **Entradas:** nada
* **Salidas:** devuelve el primer dato en la cola, el primero que ingresó. Si esta vacía lanza una excepción.
* **Poscondición:** si la cola no esta vacía, devuelve el dato y lo elimina de la cola.

1. Indique las entradas, salidas y poscondición del método Peek de la clase Queue<T>.

* **Entradas:** nada
* **Salidas:**  el primer dato que entró a la cola. Si la cola esta vacía lanza una excepción.
* **Poscondición:** Los datos permanecen sin cambio.

## Parte 2: Programe su propia *Cola*

Lectura:

* Secciones: *Implementations of Queues, Definitions, Queue Operations*, del libro *Data Structures and Program Design in C++*, páginas 84 – 88.

Programe su propia clase Cola, utilizando una interfaz similar a la que se usó en el programa donde usted escribió su propia clase Pila.