

# Práctico 8

## Implementación de Filas

NOTA: Los ejercicios deberán entregarse completos, siguiendo los criterios aconsejados por la cátedra y con los controles adecuados.

### Ejercicio 1

En la librería de FilaA vista en la cátedra implemente los operadores << (flujo de salida para imprimir la fila por pantalla) y los operadores de comparación == y !=.

### Ejercicio 2

En la librería de FilaA vista en la cátedra implemente el operador << para permitir el ingreso de elementos a la Fila. La librería debe permitir que el siguiente código se ejecute y muestre la salida indicada.

Código	Salida por consola
<pre>std::cout &lt;&lt; "FilaA" &lt;&lt; std::endl; FilaA&lt;int&gt; fila; fila &lt;&lt; 1; fila &lt;&lt; 2; std::cout &lt;&lt; fila &lt;&lt; std::endl; fila &lt;&lt; 4 &lt;&lt; 5 &lt;&lt; 6 &lt;&lt; 7; std::cout &lt;&lt; fila &lt;&lt; std::endl; std::cout &lt;&lt; fila.recuperarFrente() &lt;&lt; std::endl;</pre>	<pre>FilaA 1, 2 1, 2, 4, 5, 6, 7 1</pre>

### Ejercicio 3

En la librería FilaA vista en la cátedra implemente las siguientes operaciones básicas:

```
/**
 * @brief Retorna el elemento que está al final de la fila
 * @return T
 */
T recuperarUltimo()

/**
 * @brief Quita los n primeros elementos de la fila, en caso de que n sea
 * mayor a la cantidad de elementos la fila queda vacía.
 * @param n
 */
void quitarN(unsigned int n)

/**
```

---

```
* @brief Pone el elemento x al frente de la fila
* @param x
*/
void ponerAlFrente(T x)

/**
* @brief Invierte la fila
*/
void invertir()
```

Es un requerimiento no funcional optimizar la complejidad y el consumo de memoria en los cuatro métodos.

## Ejercicio 4

Utilizando el tipo de dato `queue` de [STL](#) implementar una función que muestre los elementos de una fila.

## Ejercicio 5

Utilizando el tipo de dato `queue` de [STL](#) implementar una función que inserte un elemento X en el frente de la fila.