

## Proyecto Programado #1 – Valor 20%

### Objetivos

- Investigar acerca de la aplicación de Heaps y Colas de Prioridad
- Implementar estructuras de datos dinámicas (lista doblemente enlazada)
- Diseñar una interfaz adecuada para interactuar con la Cola de Prioridad
- Implementar una Cola de Prioridad utilizando estructuras de datos dinámicas basado en un Heap
- Mantener la complejidad algorítmica de las funciones en una Cola de Prioridad/Heap en niveles óptimos

### Instrucciones

En este proyecto debe implementar un **TDA Cola de Prioridad** basada en un **Heap**, utilizando una **lista doblemente enlazada** como estructura subyacente. Es decir, el Heap se implementará mediante una lista doblemente enlazada. Por lo tanto, el proyecto deberá tener los siguientes componentes:

Interfaz e implementación de la Cola de Prioridad: Deberá implementar los métodos necesarios para que el usuario utilice el **TDA Cola de Prioridad** de manera adecuada. La implementación se desarrollará por medio de un **Heap** (el cual quedará oculto al usuario). Será su trabajo identificar cuáles son los métodos necesarios.

Implementación del Heap: El Heap se construirá a partir de una **lista doblemente enlazada** y deberá tener como mínimo los siguientes métodos:

- **Insertar:** Inserta un nuevo elemento dentro del Heap
- **Eliminar:** Elimina un elemento del Heap
- **Heapify:** Mantiene la propiedad de Heap de la estructura
- **CrearHeap:** Recibe una lista doblemente enlazada y crea un Heap a partir de ella

Puede agregar métodos adicionales para el manejo del Heap si lo considera necesario.

Para probar la Cola de Prioridad, implemente un pequeño programa donde se proporcione una solución a un problema donde resulte adecuado el uso de la Cola de Prioridad. Puede seleccionar cualquier problema.

### Características del Programa

- El programa debe ser desarrollado en C++ utilizando Visual Studio Community. 2017-2019
- La única interfaz disponible para el usuario es el de la Cola de Prioridad. El usuario no deberá tener acceso a los detalles de implementación de la misma, es decir, no deberán interactuar directamente ni con el **Heap** ni con la **Lista Doblemente Enlazada**.
- Los algoritmos utilizados para la solución del problema deberán mantener una complejidad óptima.

- Tome en cuenta que se puede crear una Cola de Prioridad a partir de otra, así que implemente los constructores de copia pertinentes para tal fin. Además, Deberá incluir un constructor adicional para crear una Cola de Prioridad a partir de una lista doblemente enlazada.
- En el momento de instanciar una Cola de Prioridad, se podrá determinar si se trabajará con elementos máximos (mayores – mayor prioridad) o mínimos (menores, menor prioridad). Tome en cuenta de que si no se indica en la llamada el tipo de ordenamiento, entonces se supondrá por default una Cola de Máximos.

**Tip:** Para esto agregue un parámetro default en los constructores de la clase que indique el ordenamiento.

- La Cola de Prioridad debe construirse a partir de plantillas (Templates) para que soporte cualquier objeto definido por el usuario. Tome en cuenta que será necesario comparar objetos para determinar el orden en el que deben insertarse en la Cola. Para ese fin, se recomienda lo siguiente:

a) Implemente un método dentro de la clase Cola de Prioridad que compare dos objetos T, y retorne True/False si el primer parámetro es mayor que el segundo. Por ejemplo:

```
template <typename T>
bool compara(T& obj_a, T& obj_b){
    return (obj_a > obj_b) ? true : false;
}
```

// En este ejemplo la función retorna true si el obj\_a es mayor que obj\_b

b) Para que el método **compara** funcione adecuadamente, debe sobrecargar los operadores '>', '<' y '==' dentro de las clases proporcionadas por el usuario para los objetos que se insertarán en la cola (depende de cómo realice la implementación puede ser que requiera sobrecargar uno, dos o los tres operadores).

### Observaciones

- La tarea debe ser realizada en grupos de tres personas como máximo.
- Se debe entregar el proyecto con el código fuente, ejecutables y librerías necesarias para su ejecución así como cualquier otro archivo que se requiera.
- La fecha límite para la entrega del proyecto es el día Domingo 24 de octubre de 2021.
- En caso de copia (dos proyectos o más con mucha similitud que lo demuestren) o plagio (códigos descargados de Internet, libros o cualquier otro material), la nota del proyecto es automáticamente 0.
- Debe indicar todos los recursos consultados para la elaboración del proyecto en la documentación dentro del código fuente (enlaces, libros, etc.).