

# Tercer taller obligatorio – Conjunto sobre ABB

## Consigna

En este taller deben implementar un *conjunto sobre árbol binario de búsqueda*.

Para resolver el taller cuentan con dos archivos: `Conjunto.h` y `Conjunto.hpp`. En el primero deberán completar la parte privada de la clase `Conjunto`, respetando la estructura de representación de *árbol binario de búsqueda* y en el segundo deberán completar la definición de las funciones que exporta la clase.

- `Conjunto()` — Constructor por defecto de la clase conjunto. Construye un conjunto vacío.
- `~Conjunto()` — Destructor de la clase `Conjunto`.
- `void insertar(const T&)` — Inserta un elemento en el conjunto. Si este ya existe, el conjunto no se modifica.
- `bool pertenece(const T&) const` — Determina si un elemento pertenece al conjunto.
- `void remover(const T&)` — Elimina un elemento del conjunto. Si este no existe, el conjunto no se modifica.
- `const T& siguiente(const T& elem)` — Devuelve el siguiente elemento al recibido por parámetro. Se asume que el elemento recibido se encuentra en el conjunto y existe un elemento siguiente.
- `const T& minimo() const` — Devuelve el mínimo elemento del conjunto (de acuerdo con el orden  $<$  del tipo `T`). Se asume que el conjunto no es vacío.
- `const T& maximo() const` — Devuelve el máximo elemento del conjunto (de acuerdo con el orden  $<$  del tipo `T`). Se asume que el conjunto no es vacío.
- `unsigned int cardinal() const` — Devuelve el cardinal (cantidad de elementos) del conjunto.

Además, de manera opcional, pueden completar la definición del método `void mostrar(std::ostream&) const;` que sirve para mostrar el conjunto.

La implementación que realicen **no debe perder memoria**. Recomendamos utilizar **valgrind** para testear si su implementación tiene *leaks* de memoria.