

En este taller utilizaremos C++.

El objetivo de este taller es construir un índice para una tabla en memoria usando un árbol B+.

1. Cree una clase `Tupla` que describa una tupla. Su tupla debe contener los campos de la tupla. Para este ejemplo cree tuplas específicas de una tabla de interés (e.g., la tabla estudiante).
2. Cree una clase `Tabla` que describa una tabla de tuplas. Esta clase página debe permitir agregar y eliminar tuplas, para lo cual esta clase debe tener una lista de `Tuplas`. Además, su clase debe tener un atributo nombre o identificador, y un atributo índice de la clase `ArbolB`.
3. Cree la clase `ArbolB` que debe permitir
 - a) crear un `ArbolB+` vacío;
 - b) agregar elementos al árbol;
 - c) buscar elementos por llave;
 - d) borrar elementos del árbol.

Su `ArbolB` debe definirse a partir de nodos, los cuales cuentan con valores y apuntadores. Defina estos valores y apuntados como tipos generales (use templates).

4. En el atributo índice de la clase `Tabla` los tipos del `ArbolB` deben ser los tipos del valor (el tipo de la llave primaria de las tuplas) y de los apuntadores (estos son apuntadores a `Tuplas`).
5. En la clase `Tabla` implemente métodos `buscar` y `eliminar`, que permitan realizar estas operaciones usando la llave primaria de la tabla (i.e., deben recibir como argumento el valor de la llave primaria del registro a buscar/eliminar). Estos métodos deben aprovechar el índice para buscar/eliminar registros.
6. En la clase `Tabla` implemente el método `insertar` que permite insertar tuplas a la tabla. Estos métodos debe aprovechar el índice para insertar registros.
7. En la clase `Tabla` implemente un método `buscarLista` que permita buscar registros usando la llave primaria de la tabla usando la **lista** de tuplas, SIN usar el índice.
8. Pruebe su implementación desde un script principal que inserte una gran cantidad de registros a la tabla, y luego use los métodos `buscar` y `buscarLista` para buscar registros particulares.