

SMART POINTERS

Nicolás Rojas Gutiérrez
Edwin Alejandro Forero Gómez

<https://github.com/EdwinForero29/Algorithms-Project>

En esta ocasión se redefinió la forma en la cual vamos a implementar los punteros *unique_ptr* y *shared_ptr*. Se hará por medio de una clase *Smart_pointer* para luego, sea por medio de la herencia, usar algunos métodos/operadores de esta clase para los punteros y complementar con métodos propios de cada uno.

Los métodos a implementar para la clase *Smart_pointer* serán:

- Los constructores: *smart_pointer()*; *smart_pointer(datatype*)*; *smart_pointer(const smart_pointer&)*; *smart_pointer(smart_pointer&&)*
- El destructor: *~smart_pointer()*
- La sobrecarga operador de asignación: *smart_pointer& operator=(const smart_pointer&)*; *smart_pointer& operator=(smart_pointer&&)*
- La sobrecarga del operador de desreferencia/acceso: *datatype& operator*()*; *datatype*operator->()*

Se espera que para la siguiente entrega se concluya esta parte, se pueda usar para cualquier tipo de dato y haya buen manejo de excepciones.