

SMART POINTERS

Nicolás Rojas Gutiérrez
Edwin Alejandro Forero Gómez

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

Para esta entrega se logró diseñar y construir nuestra interfaz para cada uno de los punteros *unique* y *shared*.

Los métodos a implementar para el puntero inteligente *shared* son los constructores, el destructor, *reset*, *swap*, *get*, *bool* y la sobrecarga de los operadores `=`, `[]`, `->` y `*`. Por otra parte, para el puntero inteligente *unique* tenemos los siguientes métodos: los constructores, el destructor, *release*, *reset*, *swap*, *get* y la sobrecarga de los operadores `=`, `[]`, `->` y `*`.

Brevemente, el método *reset* nos ayudará a reemplazar el objeto gestionado, *swap* permitirá intercambiar los objetos gestionados, *bool* comprobará si el puntero es no nulo, *get* nos devolverá el puntero almacenado y *release* retornará el puntero y liberará la propiedad del objeto gestionado.

En cuanto a los operadores, el operador `=` su función será asignar el *shared_ptr* o *unique_ptr*, `[]` permitirá acceder al index del arreglo almacenado, y los operadores `->` y `*` desreferenciarán el puntero.

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.