

# Laboratorio 1: Clase para generar y manipular números complejos en C++

Marlon Brenes\*

El propósito de este laboratorio es realizar una práctica en la cual se construye una clase para generar y manipular números complejos. Dicha clase, para efectos prácticos, nos sirve de guía con propósito académico. Es importante resaltar que los números complejos en C++ están incluidos como parte de la biblioteca standard, mediante `std::complex`. Para efectos prácticos, es mejor aprender dicha funcionalidad en lugar de usar nuestra propia implementación. Sin embargo, dicho ejercicio nos permite poner en práctica los conceptos de la OOP.

Las tareas son las siguientes:

1. Genere tres archivos:
  - `complex.hpp`: este archivo contiene la declaración de la clase
  - `complex.cpp`: este archivo contiene la implementación de la clase
  - `main.cpp`: este archivo contiene el `main()` con el cual pondremos en uso la funcionalidad de la clase
2. Declare una clase llamada `Complex` que contiene dos atributos de tipo `double`: `real` y `imag`
3. Genere un constructor por defecto que inicializa el número complejo a  $0.0 + i0.0$
4. Genere un constructor personalizado que inicializa la parte real y compleja
5. Genere un constructor copia apropiado
6. Genere un operador de asignación apropiado
7. Genere un destructor apropiado (note que la clase no maneja recursos dinámicos de memoria)
8. Escriba un método para obtener la norma del número complejo
9. Escriba un método para conjugar el número complejo
10. Operator overloading (sobrecarga de operadores): El compilador no sabe como realizar operaciones básicas sobre los objetos de tipo `Complex`. Debemos escribir métodos para indicar cual es el comportamiento correcto al realizar operaciones básicas en objetos de tipo `Complex`, por ejemplo, adición y multiplicación. Esto se realiza mediante la sobrecarga de operadores. Sobrecargue los siguientes operadores para que cuando operen sobre objetos de tipo `Complex`, el comportamiento adecuado es obtenido:
  - Adición `+`
  - Sustracción `-`
  - Multiplicación `*`
  - División `/`
  - Hint: La sobrecarga de operadores tiene una sintaxis muy similar al operador de asignación. La diferencia es que en estos casos uno debe definir un `Complex` temporal y devolverlo en la función, en lugar de devolver `*this`. Otra diferencia importante es que en este caso uno devuelve el objeto por valor, y no una referencia como se hace con el operador de asignación
  - Técnicamente, también deberíamos sobrecargar los operadores `+=`, `-=`, `*=` y `/=`, pero no lo vamos a hacer en este laboratorio
11. Escriba una prueba de cada método en el `main`

---

\* [marlon.brenes@utoronto.ca](mailto:marlon.brenes@utoronto.ca)