

| ACTIVIDAD      | Dynamic Array Class   |
|----------------|---|
| Código         | A04   |
| Tipo           | Evaluación continua [Evaluada]  |
| Objetivo       | Elaboración de un programa para consola de Windows que permita poner en práctica los conceptos de array dinámico, puntero, referencia, clase, constructor, atributo, método, declaración, definición, header file, source file, sobrecarga de operadores y alias.               |
| Competencias   | Capacidad para crear clases con atributos, métodos y constructores, manejar punteros y memoria dinámica, entender el paso de parámetros por valor y por referencia, sobrecargar operadores y distribuir el código entre archivos en C++.  |
| Fecha límite   | 02/03/17 in situ  |
| Herramientas   | Visual Studio 2015, GitHub  |
| Palabras clave | dynamic array, pointer, reference, class, constructor, destructor, overloading operators, method, attribute, friend, definition, declaration, header, source  |
| Referencias    | <a href="#">dynamic array</a> , <a href="#">pointer</a> , <a href="#">reference</a> , <a href="#">class</a> , <a href="#">operator overloading</a> , <a href="#">iterator</a> , <a href="#">friend</a> , <a href="#">declaration/definition</a> , <a href="#">header/source</a> |

## Setup

Añadir un nuevo proyecto de tipo Win32 Console Application llamado **DynArray** a la solución **CppProjects\_NombreApellido**. Marcar la opción de **Empty Project** al crearlo en la ventana Win32 Application Wizard. Hacer **commit** para actualizar los cambios en local.

## Definición

Construir una clase que funcione como un array dinámico que gestiona su propia reserva de memoria en el Heap y puede cambiar su tamaño durante la ejecución del programa.

## Requisitos

1. Partir de la base del archivo DynArray.hh en Moodle.
2. El proyecto debe contener 3 ficheros diferentes:
  - a. **DynArray.hh**: Contiene la declaración de la clase DynArray con sus atributos y métodos. La primera instrucción del archivo debe ser **#pragma once**, una directiva del preprocesador diseñada para asegurar que el código fuente que lo invoca sea incluido una única vez.
  - b. **DynArray.cc**: Contiene la definiciones de los métodos de la clase DynArray. Debe incluir al archivo DynArray.hh para poder tener acceso a la declaración de la clase.

- c. **main.cc**: Debe incluir al archivo DynArray para poder acceder a la declaración de la clase. No es necesario incluir el archivo DynArray.cc porque en el momento de compilar el proyecto, se produce una compilación separada por cada archivo de código fuente (cpp/cc/...). La función **main()** debe cumplir como mínimo con las siguientes características:

- Declarar 2 objetos de tipo DynArray usando 2 constructores diferentes.
- Usar 1 método de capacidad.
- Usar 1 método modificador.
- Usar 1 operador relacional.
- Imprimir por consola un array usando el operador de output sobrecargado.

## Recordatorio

Todo repositorio debe tener como colaboradores a los usuarios *tonamozota* y *jsberbel*. Se deberán hacer commit y push de todos los cambios que se van aplicando a la solución.