



---

## **ESTRUCTURAS DE DATOS II**

Grado en Ingeniería Informática

CURSO 2020/21

---

---

### Práctica 1

Programación de TAD genéricos en C++

---

## Objetivos

---

- Repasar la programación en C++ mediante el entorno CodeBlocks
- Repasar la implementación de TAD en C++
- Introducir la programación en C++ de TAD genéricos y excepciones

## Duración

---

2 sesiones

## Ejercicio 1

---

Implementar en un proyecto el TAD **MultiConjunto**, instanciado a los tipos *entero* y *carácter*, en base a la siguiente interfaz:

```
template <typename T>
class Multiconjunto {
public:
    Multiconjunto ();
    // Constructor
    bool esVacio() const;
    // Comprueba si el multiconjunto es o no vacío
    int cardinalidad() const;
    // Devuelve el número de elementos
    void anade(const T& objeto);
    // Añade un objeto de tipo T al multiconjunto
    // Se permiten elementos repetidos
    void elimina(const T& objeto);
    // Elimina todas las ocurrencias del objeto
    // pasado como parámetro
    bool pertenece(const T& objeto) const;
    // Comprueba si el objeto pasado como parámetro
    // existe en el multiconjunto
private:
    T c[100];
    // Vector de almacenamiento de elementos
    int num;
    // Indica el número de elementos en el multiconjunto
};
```

Implementar en un proyecto un programa usuario que haciendo uso del TAD `Multiconjunto`, cree y manipule objetos de tipo `Multiconjunto<int>` y `Multiconjunto<char>`, realizando inserciones, borrados, etc., de enteros y caracteres.

## Ejercicio 2

---

Implementar en un proyecto el TAD **Persona**, en base a la siguiente interfaz:

```
class Persona
{
    public:
        Persona(const string& n = "", int e = 0);
        const string& getNombre() const;
        int getEdad() const;
        void setNombre(const string& n);
        void setEdad(int e);
        bool operator==(const Persona& p) const;
    private:
        string nombre;
        int edad;
};
```

Instanciar la clase `Multiconjunto` del ejercicio 1 a esta clase, de forma que en el usuario pueda crear y manipular objetos de tipo `Multiconjunto<Persona>`, realizando inserciones, borrados, etc., de objetos de tipo persona.

Se recuerda que debe incluirse la clase `string`:

```
#include <string>
```