

Objetivos de la práctica

Vamos a realizar un programa completo en C++ para gestionar el censo de alumnos de la Universidad. Para ello, crearemos las siguientes clases:

- *Alumno*, que contendrá dni (*string*) y nombre (*string*) de un alumno, así como un booleano que nos dirá si es egresado (ya titulado) o no (matriculado en la actualidad).
- *Titulacion*, que contendrá nombre (*string*) de una titulación, un vector con sus alumnos (*Alumno*) y el tamaño del mismo (*int*).
- *Universidad*, que contendrá un vector con las titulaciones (y por tanto sus alumnos) introducidas hasta el momento y el tamaño del mismo (*int*).
- *Menu*, que contendrá un objeto *Universidad* y será el encargado de gestionar el menú de la aplicación a través de las siguientes opciones:
 - 1.- Añadir o modificar alumno: solicitará el nombre de la titulación y los datos del alumno y lo añadirá. Si la titulación ya existe, introducirá al alumno en la misma y si no creará una nueva titulación con dicho alumno. Un alumno puede estar en dos titulaciones distintas pero no dos veces en la misma titulación (lo comprobaremos a través del dni). Si el alumno ya estaba en la titulación sustituirá los antiguos datos por los nuevos (nuevo nombre y si es egresado o no).
 - 2.- Eliminar egresados: buscará en las listas todos los alumnos egresados y los eliminará. En caso de que una titulación se quede sin alumnos, la eliminará también.
 - 3.- Listar alumnos: mostrará las titulaciones y sus alumnos. Además, mostrará número total de alumnos de la Universidad y el número de alumnos por titulación.
 - 4.- Salir.

En la función *main* únicamente crearemos un objeto *Menu* y llamaremos al método del mismo encargado de mostrar el menú, recoger la opción, etc.

Itinerario de desarrollo

El ejercicio propuesto tiene muchas implementaciones posibles, por lo que se recomienda seguir el siguiente itinerario de desarrollo:

- 1) Primera implementación sencilla de las cuatro clases sin utilizar constructores, destructores ni sobrecarga de operadores. Todos los atributos serán privados y utilizaremos métodos públicos del tipo *Solicitar* y *Mostrar* para pedir al usuario y mostrar por pantalla los datos de cada clase. La clase *Universidad* contendrá un vector estático de *NTMax* (constante) elementos y un entero *N* con el número de titulaciones introducidas hasta el momento. De igual forma, cada elemento *Titulacion* tendrá un vector estático de objetos *Alumno* de *NAMax* elementos y un entero *N* con el número de alumnos introducidos hasta el momento.
- 2) Añadir los operadores de entrada/salida estándar a las clases que lo necesiten, así como los operadores adicionales que resulten de utilidad (un operador `==` en *Alumno* para comparar por DNI o un operador `+=` en *Titulacion* para añadir objetos *Alumno*, por ejemplo).
- 3) Modificar los objetos *Universidad* y *Titulacion* para que manejen vectores dinámicos (que el tamaño reservado de ambos vectores sea el mismo que el utilizado). Añadir tantos elementos (constructores, destructores y operadores) como sea necesario.
- 4) Utilizar un fichero de texto *datos.txt* en el que se cargará y guardará automáticamente la lista de alumnos y titulaciones al iniciar y terminar el programa respectivamente. Los ficheros son objetos *stream* que se utilizan de manera muy similar a otros *stream* como *cin* y *cout*. Al final de este enunciado se incluye un anexo con un ejemplo ilustrativo de cómo utilizar ficheros de texto en C++.

Entrega

La entrega de la práctica se realizará a través del Campus Virtual (dentro de la opción evaluación->controles, nunca a través de tutorías ni de cualquier otro medio) como máximo el **viernes 13 de marzo** (hasta el 13/03/2020 a las 23:59h). Constará de un solo archivo comprimido (en *zip*) con todos los archivos fuente del proyecto y un archivo de documentación (en *pdf*) que contendrá, al menos, los siguientes puntos:

Portada:

Nombre de la práctica,

Nombre, apellidos y DNI de cada uno de los componentes

Introducción:

Objetivos de la práctica (descripción de los ejercicios).

Descripción de las herramientas de programación utilizadas

Códigos fuente:

Explicación del código fuente de los ejercicios (no tiene nada que ver con los comentarios propios de cada código).

Experimentación y validación:

Explicación de las pruebas específicas que se hayan realizado para comprobar el correcto funcionamiento de la práctica

Conclusiones:

Conclusiones extraídas de la práctica.

Anexo: Ficheros en C++

A continuación se incluye un ejemplo de utilización de ficheros de texto en C++. Para utilizar un fichero es necesario abrirlo en primer lugar y cerrarlo tras su uso. En el caso de ficheros de texto se recomienda introducir cada dato en una línea para evitar ambigüedades.

```
#include <iostream>
#include <fstream> //para ficheros
#include <conio.h>
#include <stdlib.h>
using namespace std;
//guarda una agenda de teléfonos en un fichero
void GuardarTelefonos(void) {
    //abrimos el fichero de escritura (ofstream)
    //si no existe lo crea y si existe lo inicializa
    ofstream fichero("fichero.txt");
    //Pedimos cuántos contactos se desean grabar
    int N;
    cout<<"Cuántos contactos desea grabar? ";
    cin>>N;
    //Guardamos la cantidad en el fichero
    fichero<<N<<endl;
    for(int i=0;i<N;i++){
        //pedimos nombre y telefono
        cout<<"Contacto "<<i+1<<endl;
        string nombre;
        int telefono;
        cout<<"Nombre ";
        cin>>nombre;
        cout<<"Telefono ";
        cin>>telefono;
        //los guardamos en disco
        fichero<<nombre<<endl;
        fichero<<telefono<<endl;
    }
    //cerramos el fichero
    fichero.close();
    return;
}
//carga una agenda de teléfonos de un fichero y la
visualiza
void CargarTelefonos(void) {
    //abrimos el fichero de lectura (ifstream)
    ifstream fichero("fichero.txt");
    int N;
    //cargamos el número de entradas
    fichero>>N;
    for(int i=0;i<N;i++){
        string nombre;
        int telefono;
        fichero>>nombre;
        fichero>>telefono;
```

```

        cout<<"Contacto "<<i+1;
        cout<<" nombre "<<nombre;
        cout<<" telefono "<<telefono<<endl;
    }
    //cerramos el fichero
    fichero.close();
    return;
}
int main(void) {
    cout<<"Agenda telefonica"<<endl;
    cout<<"Desea cargar contactos (1) o grabarlos (2)? ";
    int op;
    cin>>op;
    switch(op) {
        case 1: CargarTelefonos(); break;
        case 2: GuardarTelefonos(); break;
        default: cout<<"Error. Opcion no valida"<<endl; break;
    }
    getche();
    return(0);
}

```