



ARQUITECTURA DE COMPUTADORES

Grado en Ingeniería Informática

**Práctica – Programación
Orientada a Objetos:
“Calificando”**

David Martínez Martínez

Índice

1. Introducción	3
2. Requisitos fundamentales	5
3. Formato de entrega	6

1. Introducción

Desarrollar una aplicación que, haciendo uso de la programación orientada a objetos, permita a un profesor:

- Asignar 1, 2 o 3 notas a los alumnos.
- Calcular la nota media según el número de notas disponibles.
- Obtener el alumno con mejor nota media, SÓLO entre aquellos que tengan 3 notas.

Para ello se debe de tener en cuenta lo siguiente:

1. Profesor y alumno deben heredar de una misma clase base y por tanto tener miembros comunes (atributos y métodos).
2. La clase base deberá llevar un constructor sobrecargado que permita crear un objeto de la clase Profesor y objetos de la clase Alumno, pasando por parámetros al constructor los campos: nombre, apellidos y DNI.
3. La aplicación no debe llevar ningún tipo de menú para la entrada de los datos. Todos los datos se crean en tiempo de compilación.

Consideraciones sobre las clases:

1. Alumno y profesor comparten datos: nombre, apellidos, dni (crear estos objetos, con estos atributos, usando un constructor sobrecargado).
2. Un profesor puede realizar estas acciones:
 - a) Añadir un nuevo alumno en su lista de alumnos.
 - b) Asignar 1, 2 o 3 notas a un alumno (sobrecarga de métodos).
 - c) Obtener la nota media de un alumno (nota media de 1, 2 o 3 notas).
 - d) Imprimir por pantalla el listado de alumnos mostrando sus datos y nota media.
 - e) Imprimir por pantalla el alumno con mejor nota (entre los alumnos con 3 notas).

3. Un alumno puede realizar estas acciones:
 - a) Añadir una nueva nota en su lista de notas.
 - b) Obtener cada una de sus notas, de su lista de notas.
 - c) Devolver el número de notas que tiene en su lista (tamaño de la lista).

Resumiendo:

El programa deberá imprimir por pantalla el listado de alumnos del profesor y mostrar el alumno con mejor nota media de entre aquellos que tengan 3 notas.

2. Requisitos fundamentales

Los requisitos FUNDAMENTALES para superar la práctica son:

- Se deben utilizar clases.
- Se debe implementar la herencia.
- Se deben usar listas (Ejemplo: <vector>).
- Se deben utilizar métodos.
- El cuerpo principal del programa debe ser este:

```
int main()  
{  
    Calificando ();  
    return 0;  
}
```

3. Formato de entrega

La práctica debe incluir todos los ficheros utilizados para su resolución. Si el proyecto ha sido desarrollado con QtCreator, se entregará la carpeta del proyecto que incluye el código fuente. Se entregará comprimido en .zip con el siguiente formato:

idusuario_practica2.zip

Donde idusuario es el usuario del correo electrónico:

idusuario@estudiantes.unileon.es

Arquitectura de Computadores

David Martínez Martínez

dmartm@unileon.es