



COMPUTER ARCHITECTURE

Grado en Ingeniería Informática

**Oriented Programming Assignment:
“Assesing”**

**Dr. D. Miguel Ángel Conde González
David Martínez Martínez**

Índice

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

1. Descripción

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) Obtener cada una de sus notas, de su lista de notas.
 - b) Devolver las notas y 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. Al finalizar la impresión el programa nos pedirá el DNI del usuario y en función del tipo de objeto al que pertenezca el DNI imprimirá lo siguiente:

- Si el DNI no existe imprimirá una alerta y volverá a solicitar un nuevo DNI
- Si el DNI pertenece a un profesor, la aplicación imprimirá los datos del profesor, su lista de alumnos (incluyendo nombre, apellidos, DNI, notas obtenidas en los exámenes y nota media), y el mejor alumno (de entre los que tienen 3 notas).
- Si el DNI pertenece a un alumno, la aplicación imprimirá los datos del alumno, las notas obtenidas y su nota media.

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 es individual.
- La entrega deberá realizarse a través de agora.unileon.es
- El código fuente se entregará en un único archivo comprimido.
- **Fecha límite de entrega:** 2 de marzo de 2020.