

Título del ejercicio: "Sistema de Gestión de Estudiantes y Calificaciones"

## Enunciado

Implementa un sistema de gestión de estudiantes y calificaciones en C# utilizando programación orientada a objetos. El sistema debe incluir las siguientes clases: **Persona**, **DNI**, **Estudiante** y **Calificacion**. A continuación, se detallan los requisitos para cada clase:

### Clase Persona

1. Implementa la clase **Persona** con las siguientes propiedades:
  - **Nombre** (string, solo lectura)
  - **Edad** (int, solo lectura)
  - **\_dni** (privado, de tipo DNI)
2. Crea dos constructores:
  - Uno que reciba edad, número de DNI y letra de DNI.
  - Otro que reciba nombre, edad, número de DNI y letra de DNI.  
(si no se recibe nombre, por defecto será "*Desconocido*")
3. Implementa una propiedad **Dni** de solo lectura que devuelva el DNI como string.
4. Sobrescribe los métodos **Equals**, **GetHashCode** y **ToString**.

### Clase DNI

1. Implementa la clase **DNI** con las siguientes propiedades:
  - **nDni** (long, solo lectura)
  - **lDni** (char, solo lectura)
2. Crea un constructor que valide la letra del DNI utilizando el método privado **LetraValida**.
3. Implementa el método privado **LetraValida** que calcule la letra correcta del DNI.
4. Sobrescribe los métodos **ToString**, **Equals** y **GetHashCode**.

### Clase Estudiante

1. Implementa la clase **Estudiante** que herede de **Persona** e implemente la interfaz **IComparable<Estudiante>**.
2. Añade una lista privada de **Calificacion** llamada **calificaciones**.
3. Crea dos constructores que llamen a los constructores de la clase base.
4. Implementa el método **Calificar** que añada una nueva calificación a la lista.

5. Implementa el método `NotaMedia` que calcule y devuelva la nota media del estudiante.
6. Implementa el método `CompareTo` de la interfaz `Comparable<Estudiante>` para comparar estudiantes por su nota media.
7. Sobrescribe el método `ToString` para mostrar la información del estudiante y sus calificaciones.

## Clase Calificacion

1. Implementa la clase `Calificacion` con las siguientes propiedades:
  - `Asignatura` (string, solo lectura)
  - `Nota` (float, solo lectura)
2. Crea un constructor que valide que la nota esté entre 0 y 10.
3. Sobrescribe el método `ToString` para mostrar la información de la calificación.

## Tareas adicionales

1. Crea una clase `Program` con un método `Main` que demuestre el funcionamiento de todas las clases.
2. En el método `Main`, crea varios objetos `Estudiante`, añade calificaciones y muestra sus datos.
3. Crea una lista de estudiantes, ordénala por nota media y muestra el resultado.
4. Busca un estudiante por su DNI en la lista de estudiantes.
5. Maneja las excepciones que puedan surgir al crear objetos con datos inválidos (por ejemplo, DNI incorrecto o nota fuera de rango).

## Consideraciones

- Asegúrate de que tu código siga las mejores prácticas de programación orientada a objetos.
  - Utiliza las propiedades y métodos proporcionados en las clases dadas.
  - Comenta tu código cuando sea necesario para explicar la lógica o decisiones de diseño.
  - Prueba tu implementación para asegurarte de que funciona correctamente en diferentes escenarios.
-