

Partiendo del zip con el proyecto de consola EjercicioCompletoLinq.zip y cargando listas iniciales de alumnos y clases realizar las siguientes acciones por separado y en métodos separados.

- Enunciado 1. Sacar un listado de alumnos en una entidad de salida que contenga el Nombre de la entidad Clase (**pista join de la lista alumnos con lista clases**). Por lo tanto se deberá crear la clase de salida correspondiente *AlumnoExtendido*:

```
NombreAlumno
FechaDeNacimientoAlumno
NotaMediaAlumno
NombreClase
```

Esta query la vamos a realizar en una clase separada que se llame *MetodosAlumnos* dentro de una función *GetAlumnosJoin* que devolverá el tipo *List<AlumnoExtendido>*. Deberá tener una interfaz *IMetodosAlumnos* que tenga el contrato de la clase. Desde Program.cs llamaremos a ese método *GetAlumnosJoin* y pintaremos el resultado por pantalla.

- Enunciado 2. Sacar un listado de alumnos de tipo *AlumnoExtendido* en una entidad de salida que contenga el Nombre de la entidad clase y la nota media del alumno (**pista join con lista clases y average sobre el campo notas**).

LINQ | ¿Cómo encontrar el promedio de la secuencia dada? – Acervo Lima

Enumerable.Average Método (System.Linq) | Microsoft Docs

Se puede hacer sobre el mismo *GetAlumnosJoin* y calcular la nota medio, o bien se puede crear una función a parte.

- Enunciado 3. Filtrar los alumnos cuyo nombre empiezan por la letra 'R' y además aplicar paginación en tamaño de pagina 2. Queremos sacar los resultados de la pagina 1 y de la página 2. Pasar la letra, el número de página y numero de registros por página en los parametros de la función *GetAlumnosJoin*.
- Enunciado 4. Añadir un filtro extra que filtre por > = nota media. (**pista -> hay que usar un let**)
- Enunciado 5. Filtrar los alumnos por fecha de nacimiento. Habrá que pasar la fecha de nacimiento por parámetro (>= fechaNacimiento).
- Enunciado 6. Dos parámetros de tipo fecha (fechaDesde y fechaHasta) y buscar todos los alumnos que tienen fecha de nacimiento entre las dos que se pasan por parámetro.
- Enunciado 7. Convertir todos los parámetros en opcionales, menos los de paginación.

- Enunciado 7. Añadir una tercera entidad, Población, cargarla con datos de prueba y añadir un campo de relación a la entidad Alumnos. Añadir la propiedad NombrePoblacion a la clase AlumnoExtendido y sacar el nombre de la población en los resultados.

```
public class Poblacion
{
    public int Id { get; set; }
    public string Nombre { get; set; }

    public static List<Poblacion> GetPoblaciones()
    {
        return new List<Poblacion>
        {
            new Poblacion
            {
                Id = 1,
                Nombre = "Madrid"
            },
            new Poblacion
            {
                Id = 2,
                Nombre = "Barcelona"
            },
            new Poblacion
            {
                Id = 3,
                Nombre = "Valencia"
            }
        };
    }
}
```

- Enunciado 8. Añadir una entidad Profesor, cargarla con datos de prueba. Crear una nueva clase MetodosProfesor con su correspondiente interfaz. Hay que implementar:
 - Una función que devuelva un listado de una nueva clase ProfesorExtendido con las propiedades del profesor además del nombre de la clase y del nombre de la población.
 - Filtrar por población
 - Añadir paginación
 - Ordenar por nombre de profesor

```

public class Profesor
{
    public string Nombre { get; set; }
    public int Clase { get; set; }
    public int PoblacionId { get; set; }
    public DateTime FechaDeNacimiento { get; set; }

    public static List<Profesor> GetProfesores()
    {
        return new List<Profesor>
        {
            new Profesor
            {
                Nombre = "Juana",
                FechaDeNacimiento = new DateTime(2002,1,1),
                Clase=3,
                PoblacionId=1
            },
            new Profesor { Nombre = "Flor", FechaDeNacimiento = new
DateTime(2003,11,21), Clase=4,PoblacionId=2 },
            new Profesor { Nombre = "Ricardo", FechaDeNacimiento = new
DateTime(2002, 12, 8), Clase=3,PoblacionId=3 },
            new Profesor { Nombre = "Aitor", FechaDeNacimiento = new
DateTime(2001,2,10), Clase=2,PoblacionId=1 },
            new Profesor { Nombre = "Antonio", FechaDeNacimiento = new
DateTime(2001,2,10), Clase=2,PoblacionId=2 },
            new Profesor { Nombre = "Rebeca", FechaDeNacimiento = new
DateTime(2002,1,1), Clase=3,PoblacionId=3 },
            new Profesor { Nombre = "Flavio", FechaDeNacimiento = new
DateTime(2001,8,28), Clase=2,PoblacionId=2 },
            new Profesor { Nombre = "Ruben", FechaDeNacimiento = new
DateTime(2006,8,28), Clase=2,PoblacionId=1 }
        };
    }
}

```