

Seminario de Lenguajes (.NET)
Práctica 11

1) Utilizando el método **Range** de la clase **System.Linq.Enumerable** y los métodos de **LINQ** que sean necesarios, obtener:

- a) Lista con todos los múltiplos de 5 entre 100 y 200
- b) Lista con todos los números primos menores que 100
- c) Lista con las potencias de 2, desde 2^0 a 2^{10}
- d) La suma y el promedio de los valores de la lista anterior
- e) Lista de todos los n^2 que terminan con el dígito 6, para n entre 1 y 20
- f) Lista con los nombres de los días de la semana en inglés que contengan una letra 'u'
(tip: utilizar el enumerativo **DayOfWeek**)

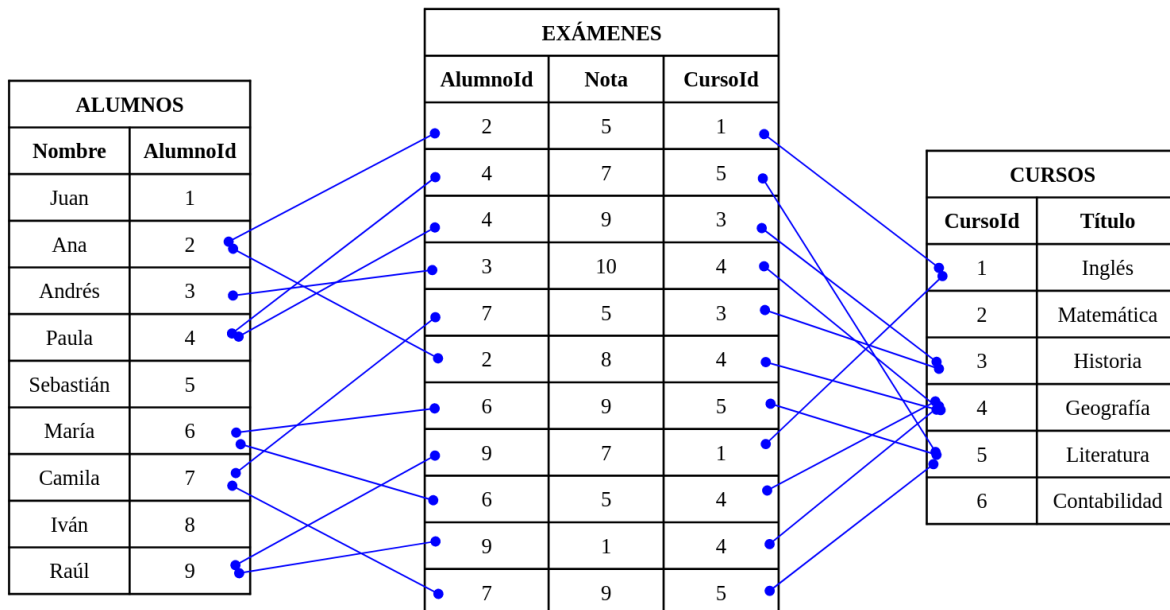
2) Listar por consola la cantidad de veces que se repiten los elementos de un vector de enteros. Ordenar por cantidad de repeticiones. Completar el siguiente código para que la salida por consola sea la indicada

```
int[] vector = new int[] { 1, 3, 4, 5, 9, 4, 3, 4, 5, 1, 1, 4, 9, 4, 3, 1 };  
vector.GroupBy(n => n)  
    // . . . completar aquí las líneas que faltan usando fluent API  
    .ForEach(obj => Console.WriteLine(obj));
```

Salida por consola

```
{ Numero = 5, Cantidad = 2 }  
{ Numero = 9, Cantidad = 2 }  
{ Numero = 3, Cantidad = 3 }  
{ Numero = 1, Cantidad = 4 }  
{ Numero = 4, Cantidad = 5 }
```

3) Supongamos que tenemos la siguiente información sobre alumnos, cursos y exámenes estructurada de esta manera:



Podemos ver, por ejemplo, que Ana sacó 5 en un examen de Inglés y 8 en un examen de Geografía, que Juan no rindió ningún examen y que nadie rindió examen en el curso de Matemáticas.

Definir las clases **Alumno**, **Examen** y **Curso**. Establecer por código las listas **alumnos** (de tipo **List<Alumno>**), **exámenes** (de tipo **List<Examen>**) y **cursos** (de tipo **List<Curso>**) con los datos que se muestran en la imagen anterior y resolver utilizando LINQ:

- Obtener el listado con los nombres de los alumnos que rindieron al menos un examen, ordenado alfabéticamente (tip: puede utilizarse el método de extensión **Distinct()** para obtener una secuencia de elementos no repetidos). La salida debería ser:

Salida por consola

```
Ana
Andrés
Camila
María
Paula
Raúl
```

- Obtener el listado con los cursos donde se hayan rendido exámenes. Se debe listar el título del curso junto con la cantidad de exámenes. El listado debe ordenarse por cantidad de exámenes. La salida debería ser:

Salida por consola

```
{ Título = Inglés, Cantidad = 2 }
{ Título = Historia, Cantidad = 2 }
{ Título = Literatura, Cantidad = 3 }
{ Título = Geografía, Cantidad = 4 }
```

- Obtener el listado con los alumnos que hayan rendido al menos un examen informando el nombre del alumno, el título del curso y la nota del examen. La salida debería ser:

Salida por consola

```
{ Alumno = Ana, Curso = Inglés, Nota = 5 }  
{ Alumno = Ana, Curso = Geografía, Nota = 8 }  
{ Alumno = Andrés, Curso = Geografía, Nota = 10 }  
{ Alumno = Paula, Curso = Literatura, Nota = 7 }  
{ Alumno = Paula, Curso = Historia, Nota = 9 }  
{ Alumno = María, Curso = Literatura, Nota = 9 }  
{ Alumno = María, Curso = Geografía, Nota = 5 }  
{ Alumno = Camila, Curso = Historia, Nota = 5 }  
{ Alumno = Camila, Curso = Literatura, Nota = 9 }  
{ Alumno = Raúl, Curso = Inglés, Nota = 7 }  
{ Alumno = Raúl, Curso = Geografía, Nota = 1 }
```

- d) Filtrar el listado del punto anterior para mostrar sólo los casos aprobados (nota ≥ 6).
- e) Obtener el listado con los nombres de los alumnos que no han rendido ningún examen.
- f) Obtener el listado de los alumnos que hayan rendido algún examen junto con el promedio de todos sus exámenes. La salida debería ser:

Salida por consola

```
{ Alumno = Ana, Promedio = 6,5 }  
{ Alumno = Andrés, Promedio = 10 }  
{ Alumno = Paula, Promedio = 8 }  
{ Alumno = María, Promedio = 7 }  
{ Alumno = Camila, Promedio = 7 }  
{ Alumno = Raúl, Promedio = 4 }
```

4. Se debe realizar un programa que permita las operaciones CRUD (Create, Read, Update, Delete) para una casa de alquiler de juegos.
- Para esto, definir las siguientes tablas desde DB Browser o bien utilizar Code First para generar la base de datos como se vio en la teoría:

Tabla

Cientes

▼ Avanzado

Campos Restricciones

➕ Añadir ➖ Eliminar ⇄ Mover al principio ▲ Mover hacia arriba ▼ Mover hacia abajo ⇄ Mover al final



Nombre	Tipo	NN	PK	AI	U	Por defecto	Check
Id	INTEGER	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
DNI	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
ApellidoYNombre	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Direccion	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Mail	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Telefono	TEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Tabla

Juegos

▼ Avanzado

Campos Restricciones

 Añadir
  Eliminar
 ⇄ Mover al principio
 ▲ Mover hacia arriba
 ▼ Mover hacia abajo
 ⇄ Mover al final



Nombre	Tipo	NN	PK	AI	U	Por defecto	Check
Id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Nombre	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Descripcion	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Estado	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
PrecioPorDia	REAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Tabla

Alquileres

▼ Avanzado

Campos Restricciones

 Añadir
  Eliminar
 ⇄ Mover al principio
 ▲ Mover hacia arriba
 ▼ Mover hacia abajo
 ⇄ Mover al final

Nombre	Tipo	NN	PK	AI	U	Por defecto	Check
Id	INTEGER	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
IdCliente	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
IdJuego	INTEGER	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
Fecha	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
FechaTentativaDevolucion	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
FechaDevolucion	TEXT	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
CostoTotal	REAL	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Nota: El tipo DateTime no existe en SQLite. Para guardar fechas se puede utilizar TEXT REAL o INTEGER.

Codificar un programa que dado el siguiente código produzca la salida en consola indicada:

```

var juego1 = new Juegos()
{
    Nombre = "Cama Elastica",
    Descripcion = "Medida de 2 x 2",
    Estado = "Bueno",
    PrecioPorDia = 1000
};
var juego2 = new Juegos()
{
    Nombre = "Castillo",
    Descripcion = "Hasta 10 personas",
    Estado = "Nuevo",
    PrecioPorDia = 1200
};
AgregarJuego(juego1);
AgregarJuego(juego2);
ListarJuegos();

var cliente1 = new Clientes()
{
    DNI = "20569784",
    ApellidoYNombre = "Perez, Juan",
    Direccion = "48 e/ 5 y 6 N°520"
};
var cliente2 = new Clientes()
{
    DNI = "10569784",
    ApellidoYNombre = "Gonzalez, Alejandra",
    Direccion = "25 e/ 9 y 10 N°520",
    Mail = "gale@gmail.com",
    Telefono = "221-15-569874"
};
AgregarCliente(cliente1);
AgregarCliente(cliente2);
ListarClientes();

var alquiler1 = new Alquileres()
{
    IdCliente = 1,
    IdJuego = 1,
    Fecha = DateTime.Now
};
var alquiler2 = new Alquileres()
{
    IdCliente = 1,
    IdJuego = 2,
    Fecha = DateTime.Now
};
AgregarAlquiler(alquiler1);
AgregarAlquiler(alquiler2);
ListarAlquileres();

ModificarCliente("10569784", "Gonzalez, Alejandra", "52 e/ 9 y 10 N°520",
"gale@gmail.com", "221-15-569874");
ListarClientes();
EliminarCliente("10569784");
ListarClientes();

ModificarJuego(1, "Cama Elastica", "Medida de 2 x 2", "Roto", 1500);
ListarJuegos();

ModificarAlquiler(1, 1562.25, new DateTime(2021 / 11 / 12));
ListarAlquileres();

```

Salida por consola:

```
-- Se agregó el juego "Cama Elastica" al que se le asignó el Id 1
-- Se agregó el juego "Castillo" al que se le asignó el Id 2

-- Listado de Juegos --
{ Id = 1, Nombre = Cama Elastica, Descripción = Medida de 2 x 2, Estado = Bueno,
PrecioPorDia = 1000 }
{ Id = 2, Nombre = Castillo, Descripción = Hasta 10 personas, Estado = Nuevo,
PrecioPorDia = 1200 }

-- Se agregó el cliente "Perez, Juan" al que se le asignó el Id 1
-- Se agregó el cliente "Gonzalez, Alejandra" al que se le asignó el Id 2

-- Listado de Clientes --
{ Id = 1, DNI = 20569784, Nombre = Perez, Juan, Dirección = 48 e/ 5 y 6 N°520, Tel =
}
{ Id = 2, DNI = 10569784, Nombre = Gonzalez, Alejandra, Dirección = 25 e/ 9 y 10
N°520, Tel = 221-15-569874 }

-- Se agrego un Alquiler al que se le asignó el id 1
-- Se agrego un Alquiler al que se le asignó el id 2

-- Listado de Alquileres
{ Id = 1, Fecha = 03/11/2021, FechaDevolucion = / / , Cliente = Perez, Juan,
Juego = Cama Elastica, Monto = 0 }
{ Id = 2, Fecha = 03/11/2021, FechaDevolucion = / / , Cliente = Perez, Juan,
Juego = Castillo, Monto = 0 }

-- Se modificó cliente con dni 10569784 --

-- Listado de Clientes --
{ Id = 1, DNI = 20569784, Nombre = Perez, Juan, Dirección = 48 e/ 5 y 6 N°520, Tel =
}
{ Id = 2, DNI = 10569784, Nombre = Gonzalez, Alejandra, Dirección = 52 e/ 9 y 10
N°520, Tel = 221-15-569874 }

-- Se eliminó el cliente con DNI = 10569784 --

-- Listado de Clientes --
{ Id = 1, DNI = 20569784, Nombre = Perez, Juan, Dirección = 48 e/ 5 y 6 N°520, Tel =
}

-- Se modificó el juego con id 1 --

-- Listado de Juegos --
{ Id = 1, Nombre = Cama Elastica, Descripción = Medida de 2 x 2, Estado = Roto,
PrecioPorDia = 1500 }
{ Id = 2, Nombre = Castillo, Descripción = Hasta 10 personas, Estado = Nuevo,
PrecioPorDia = 1200 }

-- Se modificó el alquiler id 1 --

-- Listado de Alquileres
{ Id = 1, Fecha = 03/11/2021, FechaDevolucion = 12/11/2021, Cliente = Perez, Juan,
Juego = Cama Elastica, Monto = 1562.25 }
{ Id = 2, Fecha = 03/11/2021, FechaDevolucion = / / , Cliente = Perez, Juan,
Juego = Castillo, Monto = 0 }
```