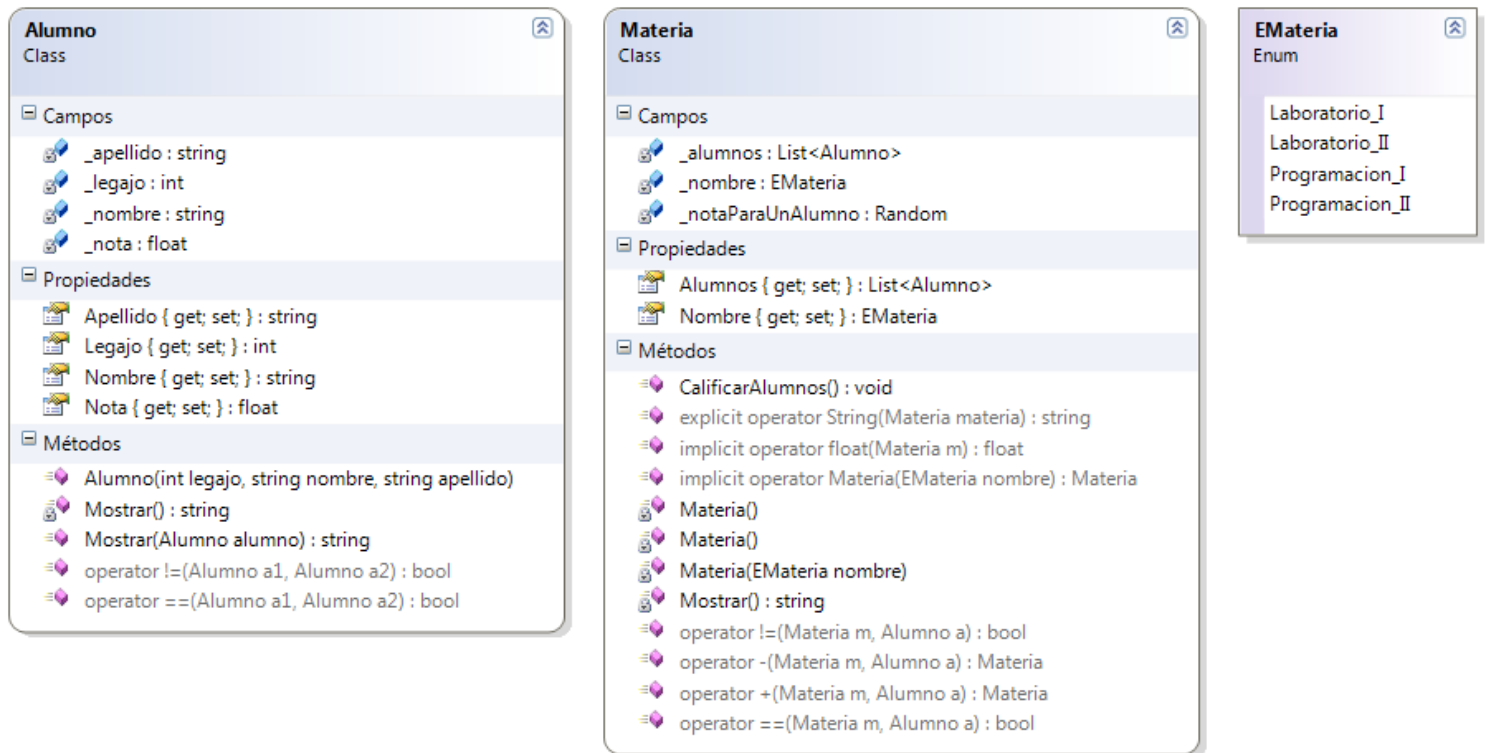


1. Generar una *Biblioteca de Clases* (Entidades) que contenga las clases Materia y Alumno.



Notas técnicas:

Alumno.

2. Generar 4 sobrecargas más al constructor, en dónde todos los atributos que se reciban tengan uso en la clase y se reutilicen la mayor cantidad de código posible.
3. El método de **instancia** `Mostrar` (*privado*) retornará en un string el contenido de sus atributos. *Ver salida de consola.*
4. El método de **clase** `Mostrar` (*público*) retornará en un string el contenido de sus atributos (reutilizar código).
5. La sobrecarga del operador `==` retornará *true*, si los legajos de los alumnos son iguales, *false*, caso contrario.

Materia.

6. Posee un constructor por defecto (*privado y de instancia*) que será el único que inicialice la lista de alumnos. El otro constructor de instancia (recibe un **EMateria**), también es *privado*.
7. El constructor de **clase**, será el encargado de inicializar al atributo de clase **notaParaUnAlumno**.

Métodos:

8. `Mostrar` (*privado*), retornará en un string el contenido de sus atributos. *Ver salida de consola.*

9. CalificarAlumnos (*público*) es el encargado de, utilizando el atributo de tipo **Random**, colocar una nota aleatoria a cada alumno de la materia.

Sobrecarga de operadores:

10. Implícito (EMateria), retornará una materia cuyo nombre sea el recibido por parámetro.
11. Implícito (Materia), retornará el promedio de todas las notas de sus alumnos.
12. Explícito (Materia), retornará un string con el contenido completo de la materia. *Ver salida de consola.*
13. Igualdad, retornará *true*, si es que el alumno se encuentra en la lista de alumnos, *false*, caso contrario.
14. Adición, retornará una **materia** con el agregado de un alumno, siempre y cuando dicho alumno **no** se encuentre en el listado. Reutilizar código.
15. Sustracción, retornará una **materia** sin el alumno, siempre y cuando el alumno se encuentre en el listado. Reutilizar código.

Agregar a la solución un proyecto de tipo Aplicación de Consola (Test) y copiar las siguientes líneas al Main.

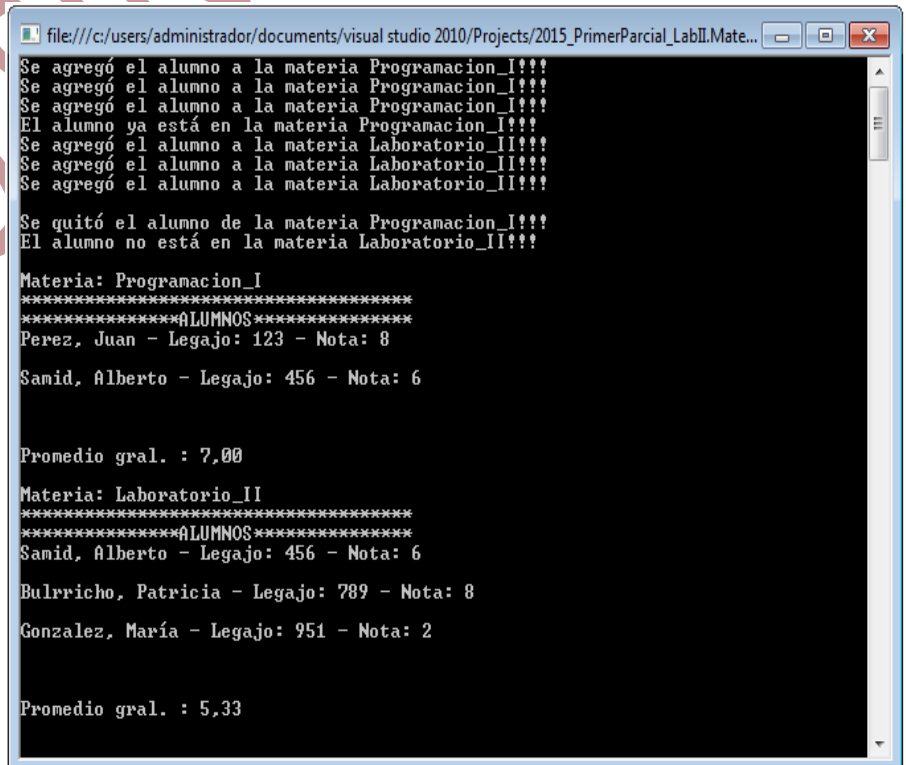
Método Main: Copiar las siguientes líneas:

```
Alumno a1 = new Alumno(123, "Juan", "Perez");
Alumno a2 = new Alumno(456, "Alberto", "Samid");
Alumno a3 = new Alumno(789, "Patricia", "Bulrricho");
Alumno a4 = a1;
Alumno a5 = new Alumno(951, "María", "Gonzalez");
```

```
Materia m1 = EMateria.Programacion_I;
Materia m2 = EMateria.Laboratorio_II;
m1 += a1;
m1 += a2;
m1 += a3;
m1 += a4;
m2 += a2;
m2 += a3;
m2 += a5;
Console.WriteLine();
m1 -= a3;
m2 -= a4;
```

```
m1.CalificarAlumnos();
m2.CalificarAlumnos();

Console.WriteLine((string)m1);
Console.WriteLine("Promedio gral. : {0:#,###.00}", (float)m1);
Console.WriteLine((string)m2);
Console.WriteLine("Promedio gral. : {0:#,###.00}", (float)m2);
Console.ReadLine();
```



```
file:///c:/users/administrador/documents/visual studio 2010/Projects/2015_PrimerParcial_LabII.Mate...
Se agregó el alumno a la materia Programacion_I!!!
Se agregó el alumno a la materia Programacion_I!!!
Se agregó el alumno a la materia Programacion_I!!!
El alumno ya está en la materia Programacion_I!!!
Se agregó el alumno a la materia Laboratorio_II!!!
Se agregó el alumno a la materia Laboratorio_II!!!
Se agregó el alumno a la materia Laboratorio_II!!!

Se quitó el alumno de la materia Programacion_I!!!
El alumno no está en la materia Laboratorio_II!!!

Materia: Programacion_I
*****ALUMNOS*****
Perez, Juan - Legajo: 123 - Nota: 8
Samid, Alberto - Legajo: 456 - Nota: 6

Promedio gral. : 7,00

Materia: Laboratorio_II
*****ALUMNOS*****
Samid, Alberto - Legajo: 456 - Nota: 6
Bulrricho, Patricia - Legajo: 789 - Nota: 8
Gonzalez, María - Legajo: 951 - Nota: 2

Promedio gral. : 5,33
```