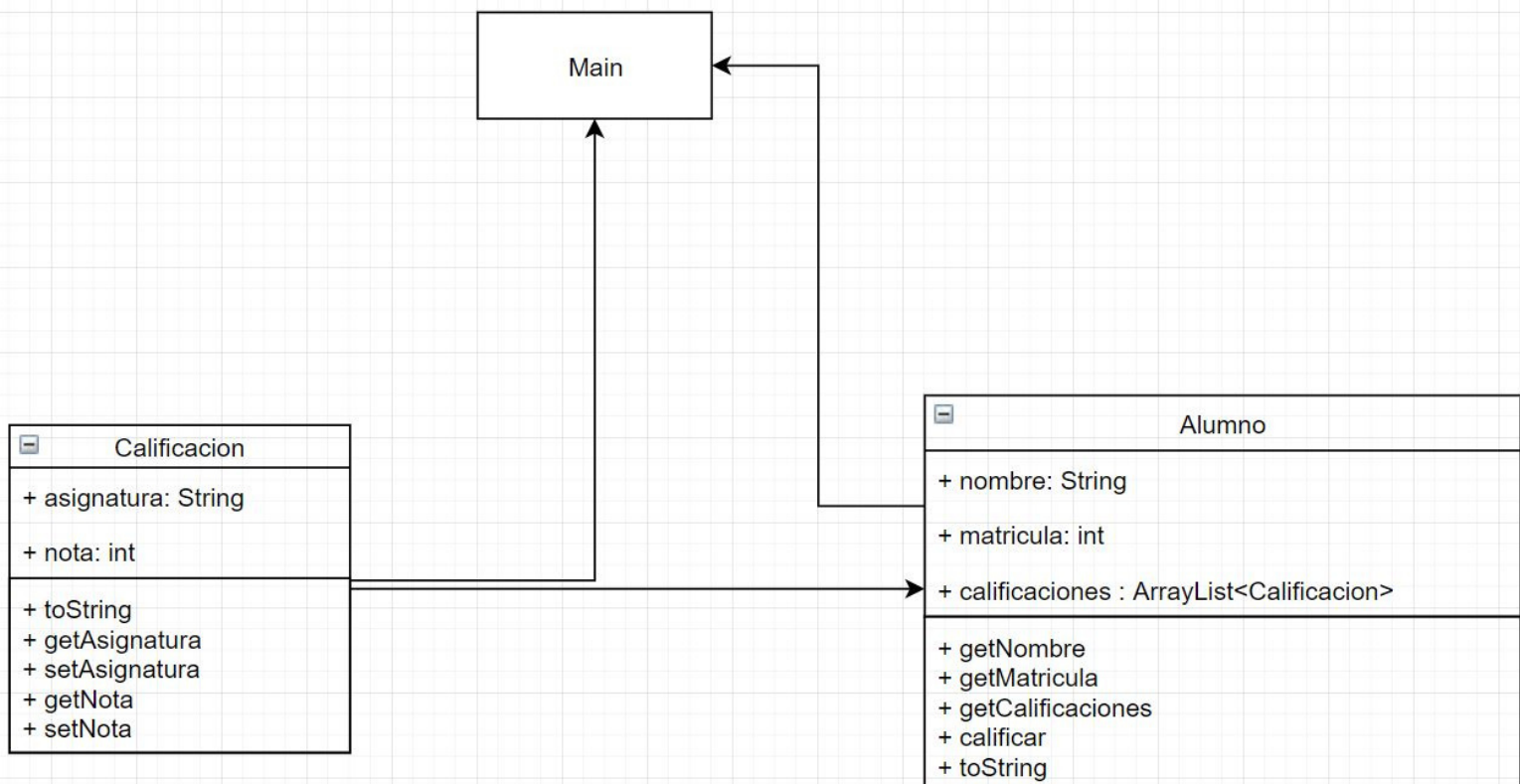


ACTIVIDAD 8. TAREA INDIVIDUAL. GESTIÓN CALIFICACIONES

Estructura:

El programa consta de 2 clases, **Calificación** y **Alumno**, la cuales serán utilizadas en la clase **main**, para ello creamos sus propiedades y sus métodos.



Metodos: Calificación

```
11 public class Calificacion {
```

```
12     //Declaro las variables
```

```
13     private String asignatura;
```

```
14     private int nota = 0; // Valor de 0 a 100
```

```
15     // Creo el constructor la clase Calificacion
```

```
16     public Calificacion(String asignatura, int nota) {
```

```
17         this.asignatura = asignatura;
```

```
18         this.nota = nota;
```

```
19     }
```

```
20     public Calificacion() {
```

```
21         this.asignatura = "";
```

```
22         this.nota = 0;
```

```
23     }
```

```
24     @Override
```

```
25     public String toString() {
```

```
26         return asignatura+nota;
```

```
27     }
```

```
28     public String getAsignatura() {
```

```
29         return asignatura;
```

```
30     }
```

```
31     public void setAsignatura(String asignatura) {
```

```
32         this.asignatura = asignatura;
```

```
33     }
```

```
34     public int getNota() {
```

```
35         return nota;
```

```
36     }
```

```
37     public void setNota(int nota) {
```

```
38         this.nota = nota;
```

```
39     }
```

```
40 }
```

Declaración de variables

Creación del constructor

Sobreescritura del método toString

**Métodos
getter/setter**

Metodos: Alumno

```
7 import java.util.ArrayList;
```

```
9 /**
10  *
11  * @author Rodri
12  */
```

```
13 public class Alumno {
```

```
14     //Declaro las variables
```

```
15     private String nombre;
```

```
16     private int matricula;
```

```
17     public ArrayList<Calificacion> calificaciones;
18     Calificacion cal;
```

Declaración de variables

```
19     // Creo el constructor la clase alumno
```

```
20     public Alumno(String nombre, int matricula) {
21         this.nombre = nombre;
22         this.matricula = matricula;
23         calificaciones=new ArrayList<Calificacion>();
24     }
25     public Alumno() {
```

Creación del constructor

```
26         this.nombre = "";
27         this.matricula = 0;
28         calificaciones=new ArrayList<Calificacion>();
29     }
30 }
```

```
31     public String getNombre() {
32         return nombre;
33     }
```

```
34     public int getMatricula() {
35         return matricula;
36     }
```

**Métodos
getter/setter**

```
37     public ArrayList<Calificacion> getCalificaciones() {
38         return calificaciones;
39     }
```

```
40     /**
41      * Este metodo sirve para asignar y almacenar la asignatura y su nota
42      * en las calificaciones de cada alumno
43      * @param asignatura tipo cadena de texto
44      * @param nota tipo numerico
45      */
```

```
46     public void calificar(String asignatura, int nota) {
47         Calificacion cal = new Calificacion(asignatura, nota);
48         calificaciones.add(cal);
49     }
```

**Método
calificar**

```
50     /**
51      * Este metodo devuelve en una cadena de texto por pantalla
52      * nombre del alumno/a+ n°matricula+calificaciones+su media
53      */
```

```
54     @Override
55     public String toString() {
56         double media=0;int suma=0;
57         for (int i=0;i<calificaciones.size();i++) {
58             suma=suma+calificaciones.get(i).getNota();
59         }
60         media=suma/ calificaciones.size();
61         return "El alumno con nombre : "+this.getNombre()+"y matricula: "+
62             this.getMatricula()+" Calificaciones: "+
63             this.calificaciones.toString()+" Nota media "+media;
64     }
65 }
```

**Sobreescritura del
método toString**

Este método sobreescribe toString para que devuelva en una cadena el nombre, la matricula las calificaciones con el array y su nota media

En este método se crea un objeto de tipo calificación que se añadirá a el ArrayList de calificaciones

Calculamos la media utilizando un for de calificaciones para recorrer el array

Main:

```
    */  
    public class EjercicioUF701 {  
  
        /**  
         * @param args the command line arguments  
         */  
        public static void main(String[] args) {  
  
            Alumno alu = new Alumno ("Paula Gonzalez ", 2347323);  
            alu.calificar("Matematicas ", 70);  
            alu.calificar("Plastica ", 80);  
            alu.calificar("Lengua ", 90);  
  
            Alumno alu1 = new Alumno ("Jesus Rodriguez ", 2364739);  
            alu1.calificar("Matematicas ", 60);  
            alu1.calificar("Plastica ", 50);  
            alu1.calificar("Lengua ", 80);  
  
            System.out.println(alu.toString());  
            System.out.println(alu1.toString());  
  
        }  
    }  
}
```

En la clase main he creado 2 alumnos (alu, alu1) a los cuales les asigno un **nombre** y n° de **matricula** cuando los creo, y con el método **calificar** añado la **asignatura** y su **nota** para que se guarden y mas abajo se muestre toda la información con el método toString en el System.out.println de los respectivos **alumnos**.

Salida de consola:

```
ant -f "C:\\Users\\Rodri\\OneDrive\\Documentos\\NetBeansProjects\\Ejercicio UF7-01" -Dnb.internal.action.name=run run  
init:  
Deleting: C:\\Users\\Rodri\\OneDrive\\Documentos\\NetBeansProjects\\Ejercicio UF7-01\\build\\built-jar.properties  
deps-jar:  
Updating property file: C:\\Users\\Rodri\\OneDrive\\Documentos\\NetBeansProjects\\Ejercicio UF7-01\\build\\built-jar.properties  
Compiling 1 source file to C:\\Users\\Rodri\\OneDrive\\Documentos\\NetBeansProjects\\Ejercicio UF7-01\\build\\classes  
compile:  
run:  
El alumno con nombre : Paula Gonzalez y matricula: 2347323 Calificaciones: [Matematicas 70, Plastica 80, Lengua 90] Nota media 80.0  
El alumno con nombre : Jesus Rodriguez y matricula: 2364739 Calificaciones: [Matematicas 60, Plastica 50, Lengua 80] Nota media 63.0  
BUILD SUCCESSFUL (total time: 0 seconds)
```