



# Proyecto Integrador

Parte 2

**Nombre:** Luis Felipe Cavazos Arias

**Matrícula:** 620204021

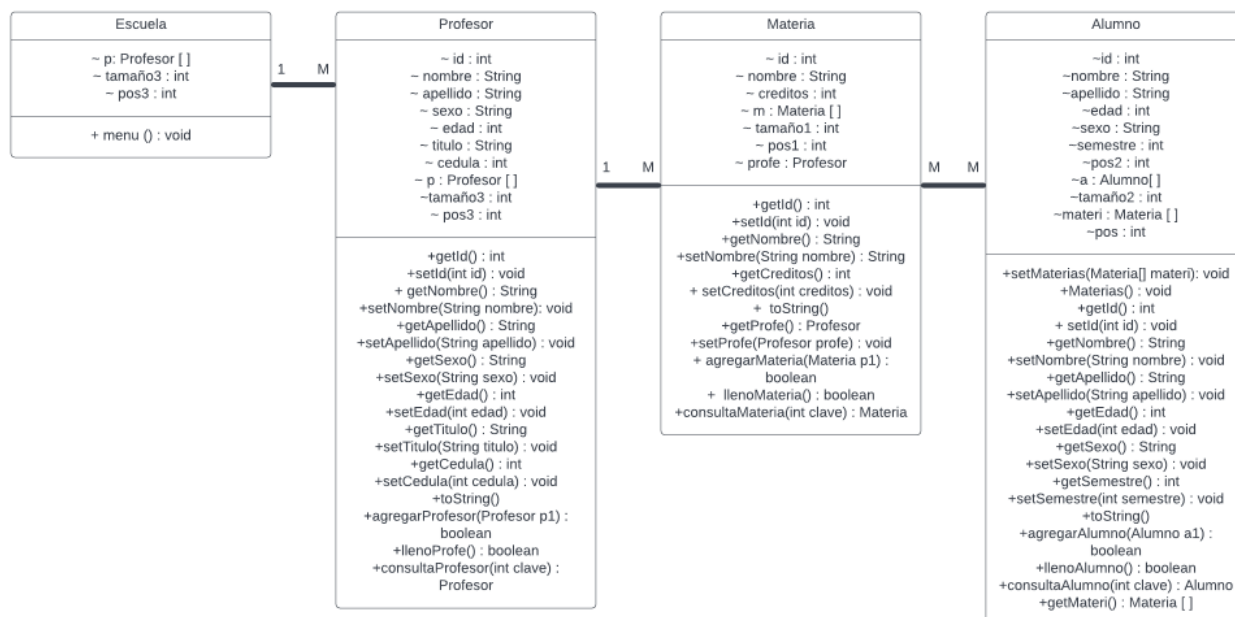
**Materia:** POO

**Profesor:** Armando Cruz Cruz

**Campus:** Villahermosa

**Fecha de entrega:** 15/11/22

## Diagrama UML



## Descripción del programa

Este programa lo que hace es simular una base de datos de una escuela, donde tienes que ingresar los profesores, luego tienes que ingresar a las materias, allí podrás seleccionar qué profesores darán cada materia, luego de eso se agregan los alumnos, ahí mismo puedes agregar qué materia llevará cada alumno.

## Capturas de pantalla

### Código main

```

1 package integrador;
2
3 import java.util.Scanner;
4
5 public class Integrador {
6
7     public static void main(String[] args) throws Exception {
8
9         Escuela es = new Escuela();
10        es.menu();
11
12    }
13 }
14
  
```

## Código clase profesor

```
1 package integrador;
2
3 public class Profesor {
4
5     private int id;
6     private String nombre;
7     private String apellido;
8     private String sexo;
9     private int edad;
10    private String titulo;
11    private int cedula;
12    private Profesor[] p;
13    private int tamaño3;
14    private int pos3;
15
16    public Profesor() {
17        id = 0;
18        nombre = "";
19        apellido = "";
20        sexo = "";
21        edad = 0;
22        titulo = "";
23        cedula = 0;
24        tamaño3 = 5;
25        p = new Profesor[tamaño3];
26        pos3 = 0;
27
28    public Profesor(int id, String nombre, String apellido, String sexo, int edad, String titulo, int cedula) throws Excepti
29    {
30        setId(id);
31        setNombre(nombre);
32        setApellido(apellido);
33        setSexo(sexo);
34        setEdad(edad);
35        setTitulo(titulo);
36        setCedula(cedula);
37    }
38
39    public int getId() {
40        return id;
41    }
42
43    public void setId(int id) throws Exception {
44        if (id >= 0) {
45            this.id = id;
46        } else {
47            throw new Exception(message: "Matricula invalida");
48        }
49    }
50
51    public String getNombre() {
52        return nombre;
53    }
54 }
```

```
56 public void setNombre(String nombre) throws Exception {
57     if (!nombre.isEmpty()) {
58         this.nombre = nombre;
59     } else {
60         throw new Exception(message: "Nombre invalido");
61     }
62 }
63
64 public String getApellido() {
65     return apellido;
66 }
67
68 public void setApellido(String apellido) throws Exception {
69     if (!apellido.isEmpty()) {
70         this.apellido = apellido;
71     } else {
72         throw new Exception(message: "Apellido invalido");
73     }
74 }
75
76 public String getSexo() {
77     return sexo;
78 }
79
80 public void setSexo(String sexo) throws Exception {
81     if (!sexo.isEmpty()) {
82         this.sexo = sexo;
83     } else {
84         throw new Exception(message: "Sexo invalido");
85     }
86 }
87
88 public int getEdad() {
89     return edad;
90 }
91
92 public void setEdad(int edad) throws Exception {
93     if (edad > 0) {
94         this.edad = edad;
95     } else {
96         throw new Exception(message: "edad invalida");
97     }
98 }
99
100 public String getTitulo() {
101     return titulo;
102 }
```

```
104 public void setTitulo(String titulo) throws Exception {
105     if (!titulo.isEmpty()) {
106         this.titulo = titulo;
107     } else {
108         throw new Exception(message: "Titulo invalido");
109     }
110 }
111
112 public int getCedula() {
113     return cedula;
114 }
115
116 public void setCedula(int cedula) throws Exception {
117     if (cedula > 0) {
118         this.cedula = cedula;
119     } else {
120         throw new Exception(message: "Celuda invaluda");
121     }
122 }
123
124 @Override
125 public String toString() {
126     String s = String.format("Format: \"%d\\t%s\\t%s\\t\\t%d\\t%s\\t%s\\t%d\"", args: id, args: nombre, args: apellido, args: edad, args: se:
127     return s;
128 }
129
```

```
130 public boolean agregarProfesor(Profesor p1) {
131     if (pos3 < tamaño3) {
132         p[pos3] = p1;
133         pos3++;
134         return true;
135     } else {
136         return false;
137     }
138 }
139
140 public boolean llenoProfe() {
141     return pos3 == tamaño3;
142 }
143
144 public Profesor consultaProfesor(int clave) {
145     Profesor profe = null;
146     if (pos3 > 0) {
147         for (int i = 0; i < pos3; i++) {
148             if (clave == p[i].getId()) {
149                 profe = p[i];
150                 break;
151             }
152         }
153     }
154     return profe;
155 }
156
```

## Código clase materia

```
1 package integrador;
2
3 public class Materia {
4
5     private int id;
6     private String nombre;
7     private int credits;
8     private Materia[] m;
9     private int tamaño1;
10    private int pos1;
11    private Profesor profe;
12
13    public Materia() {
14
15        id = 0;
16        nombre = "";
17        credits = 0;
18        tamaño1 = 3;
19        m = new Materia[tamaño1];
20        pos1 = 0;
21    }
22
23
24    public Materia(int id, String nombre, int credits, Profesor profe) throws Exception {
25        setId(id);
26        setNombre(nombre);
27        setCredits(credits);
28        setProfe(profe);
29
30
31    public int getId() {
32        return id;
33    }
34
35    public void setId(int id) throws Exception {
36        if (id >= 0) {
37            this.id = id;
38        } else {
39            throw new Exception(message: "ID invalida");
40        }
41    }
42
43    public String getNombre() {
44        return nombre;
45    }
46
47    public void setNombre(String nombre) throws Exception {
48        if (!nombre.isEmpty()) {
49            this.nombre = nombre;
50        } else {
51            throw new Exception(message: "Nombre invalido");
52        }
53    }
```

```
55 public int getCreditos() {
56     return creditos;
57 }
58
59 public void setCreditos(int creditos) throws Exception {
60     if (creditos > 0) {
61         this.creditos = creditos;
62     } else {
63         throw new Exception( message: "Creditos invalidos");
64     }
65 }
66
67 @Override
68 public String toString() {
69     String s = String.format( format: "%d\t%s\t%d", args: id, args: nombre, args: creditos);
70     return s;
71 }
72
73
74
75
76 public Profesor getProfe() {
77     return profe;
78 }
79
80 public void setProfe(Profesor profe) {
81     this.profe = profe;
82 }
83
84 public boolean agregarMateria(Materia p1) {
85     if (pos1 < tamaño1) {
86         m[pos1] = p1;
87         pos1++;
88         return true;
89     } else {
90         return false;
91     }
92 }
93
94
95 public boolean llenoMateria() {
96     return pos1 == tamaño1;
97 }
98
99 public Materia consultaMateria(int clave) {
100     Materia p = null;
101     if (pos1 > 0) {
102         for (int i = 0; i < pos1; i++) {
103             if (clave == m[i].getId()) {
104                 p = m[i];
105                 break;
106             }
107         }
108     }
109     return p;
110 }
111 }
```

## Código clase alumno

```

1  package integrador;
2
3  public class Alumno {
4
5      private int id;
6      private String nombre;
7      private String apellido;
8      private int edad;
9      private String sexo;
10     private int semestre;
11     private int pos2;
12     private Alumno[] a;
13     private int tamaño2;
14     private Materia[] materi;
15     private int pos = 0;
16
17     public Alumno() {
18         id = 0;
19         nombre = "";
20         apellido = "";
21         edad = 0;
22         sexo = "";
23         semestre = 0;
24         tamaño2=2;
25         a = new Alumno[tamaño2];
26         pos2 = 0;
27
28
29
30     public Alumno(int id, String nombre, String apellido, int edad, String sexo, int semestre, Materia[] materis) throws {
31         setId(id);
32         setNombre(nombre);
33         setApellido(apellido);
34         setEdad(edad);
35         setSexo(sexo);
36         setSemestre(semestre);
37         setMaterias(materi:materis);
38     }
39
40     public void setMaterias(Materia[] materi) {
41         this.materi = materi;
42     }
43
44     public void Materias() {
45
46         for (int i = 0; i < pos+1; i++) {
47             System.out.println(pos+materi[i].toString());
48         }
49     }
50
51     public int getId() {
52         return id;
53     }

```



```
55 public void setId(int id) throws Exception {
56     if (id >= 0) {
57         this.id = id;
58     } else {
59         throw new Exception(message: "Matricula invalida");
60     }
61 }
62
63 public String getNombre() {
64     return nombre;
65 }
66
67 public void setNombre(String nombre) throws Exception {
68     if (!nombre.isEmpty()) {
69         this.nombre = nombre;
70     } else {
71         throw new Exception(message: "Nombre invalido");
72     }
73 }
74
75 public String getApellido() {
76     return apellido;
77 }
78
79 public void setApellido(String apellido) throws Exception {
80     if (!apellido.isEmpty()) {
81         this.apellido = apellido;
82     } else {
83         throw new Exception(message: "Apellido invalido");
84     }
85 }
86
87 public int getEdad() {
88     return edad;
89 }
90
91 public void setEdad(int edad) throws Exception {
92     if (edad > 0) {
93         this.edad = edad;
94     } else {
95         throw new Exception(message: "Edad invalida");
96     }
97 }
98
99 public String getSexo() {
100     return sexo;
101 }
```

```
103 public void setSexo(String sexo) throws Exception {
104     if (!sexo.isEmpty()) {
105         this.sexo = sexo;
106     } else {
107         throw new Exception(message: "Sexo invalido");
108     }
109 }
110
111 public int getSemestre() {
112     return semestre;
113 }
114
115 public void setSemestre(int semestre) throws Exception {
116     if (semestre > 0) {
117         this.semestre = semestre;
118     } else {
119         throw new Exception(message: "Semestre invalido");
120     }
121 }
122 }
```

```
124 @Override
125 public boolean equals(Object obj) {
126     if ((obj instanceof Alumno)) {
127         Alumno tmp = (Alumno) obj;
128         if (this.id == tmp.id && this.nombre.equalsIgnoreCase(anotherString: tmp.nombre)
129             && this.apellido.equalsIgnoreCase(anotherString: tmp.apellido)
130             && this.edad == tmp.edad && this.sexo.equalsIgnoreCase(anotherString: tmp.sexo) && this.semestre == tmp.semestre)
131             return true;
132         } else {
133             return false;
134         }
135     } else {
136         return false;
137     }
138 }
139
140 @Override
141 public String toString() {
142     String s = String.format(format: "%d\t%s\t%s\t\t%d\t%s\t%d", args: id, args: nombre, args: apellido, args: edad, args: sexo,
143                             args: semestre);
144     return s;
145 }
146 }
```

```
149 public boolean agregarAlumno(Alumno a1) {
150     if (pos2 < tamaño2) {
151         a[pos2] = a1;
152         pos2++;
153         return true;
154     } else {
155         return false;
156     }
157 }
158
159 public boolean llenoAlumno() {
160     return pos2 == tamaño2;
161 }
162
163 public Alumno consultaAlumno(int clave) {
164     Alumno p = null;
165     if (pos2 > 0) {
166         for (int i = 0; i < pos2; i++) {
167             if (clave == a[i].getId()) {
168                 p = a[i];
169                 break;
170             }
171         }
172     }
173     return p;
174 }
```

```
175 public Materia[] getMateri() {
176     return materi;
177 }
178
179
180
181 }
182
```

## Capturas corriendo

```
Output - Integrador (run) X
run:
Elige la opcion que quieras hacer: Opcion: 1
0. Atras
1. Agregar
2. Imprimir
Opcion: 1
Elige la opcion a agregar:
0. Salir
1. Profesor
2. Materia
3. Alumno
Opcion:
Ingresa la matricula: 1
Ingresa el nombre: Armando
Ingresa el apellido: Cruz
Ingresa el sexo: Hombre
Ingresa la edad: 30
Ingresa el titulo: SC
Ingresa la cedula: 2020
Profesor agregado!
```

```
Output - Integrador (run) X
Elige la opcion a agregar:
0. Salir
1. Profesor
2. Materia
3. Alumno
Opcion: 2
Ingresa la ID de la materia: 1
Ingresa su nombre: POO
Ingresa el numero de creditos: 20
Ingresa la matricula del profesor que la impartira: 1
Materia agregada!
```

```
Output - Integrador (run) X
Ingresa matricula: 1
Ingresa su nombre: Luis
Ingresa su apellido: Cavazos
Ingresa su edad: 19
Ingresa su sexo: Hombre
Ingresa su semestre: 3
Ingresa las materias, cuando termine ingrese 0
1
Esta materia fue agregada
0
Alumno agregado!
```

```
Output - Integrador (run) X
Elige la opcion a imprimir:
1. Alumno
2. Profesor
3. Materia
Opcion: 1

Ingrese la matricula del alumno: 1
ID      Nombre  Apellido      Edad   Sexo   Semestre
1       Luis    Cavazos       19     Hombre 3

Materias:
ID      Nombre  Creditos
01      POO      20
```

Elige la opcion a imprimir:

- 1. Alumno
- 2. Profesor
- 3. Materia

Opcion: 2

Ingrese la matricula: 1

ID	Nombre	Apellido	Edad	Sexo	Titulo	Cedula
1	Armando	Cruz	30	Hombre	SC	2020

```
Elige la opcion a imprimir:
1. Alumno
2. Profesor
3. Materia
Opcion: 3

Ingrese la ID: 1
ID      Nombre  Creditos
1       POO      20

ID      Nombre  Apellido      Edad   Sexo   Titulo  Cedula
1       Armando Cruz    30     Hombre  SC      2020
```

Output - Integrador (run) X



1. Alumno  
2. Profesor  
3. Materia  
Opcion: 3

Ingrese la ID: 1

ID	Nombre	Creditos
1	POO	20

ID	Nombre	Apellido	Edad	Sexo	Titulo	Cedula
1	Armando	Cruz	30	Hombre	SC	2020

Elige la opcion que quieras hacer:

0. Atras  
1. Agregar  
2. Imprimir  
Opcion: 0

Hasta luego...

BUILD SUCCESSFUL (total time: 3 minutes 1 second)