

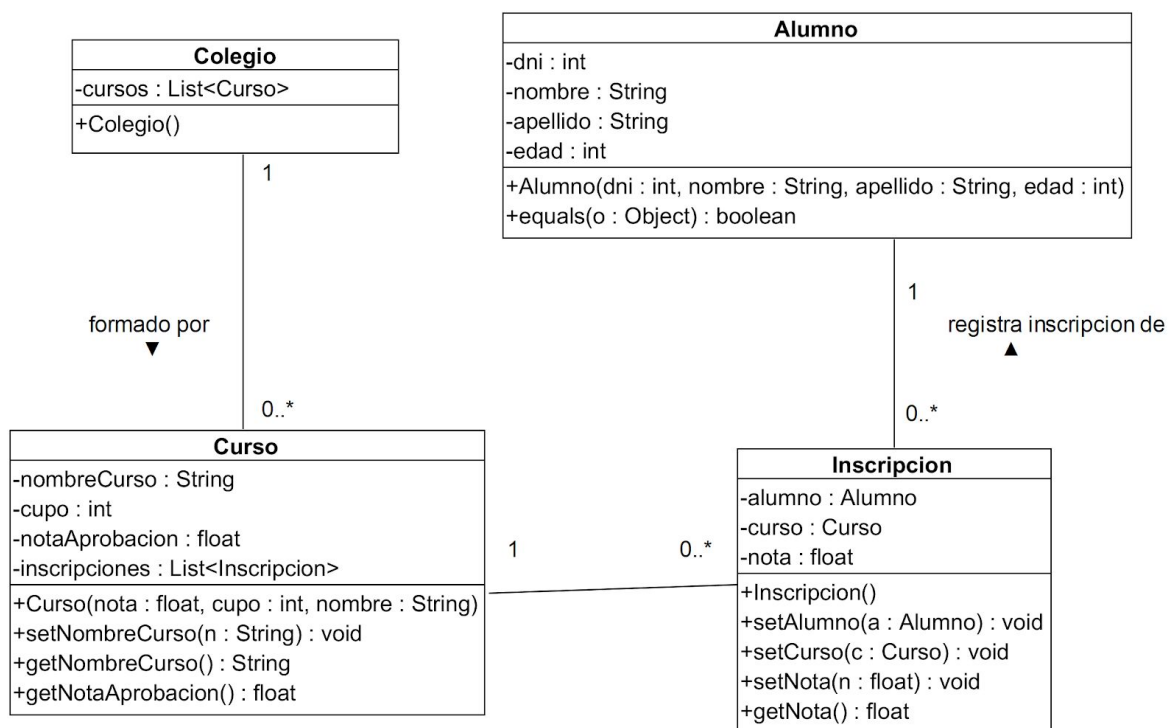
## Examen 111MIL - Inscripción a Curso 111MIL - Solución

Adolfo, el director del Colegio Secundario N° 1, ha abierto el período de inscripción para la realización de los cursos del Programa 111 Mil. En dicho colegio se van a dictar 4 cursos: Técnicas de Programación, Programación Orientada a Objetos, Base de Datos y Desarrollo de Software. El período de inscripción a los cursos va desde el 01/03/2018 al 31/03/2018 en la secretaría del colegio. Adolfo desea contar con un sistema que le permita realizar la siguiente funcionalidad: consultar cupo para un curso, inscribir un alumno a un curso, consultar fecha de inscripción o de exámenes, cargar y consultar notas, y obtener la cantidad de alumnos en un curso. Como Adolfo sabe que Esteban, profesor de Informática, está trabajando en el programa 111 Mil, le pide si le puede recomendar algún programador que se haya certificado en el programa, para ayudarlo a resolver los problemas que tiene para manejar la inscripción de alumnos a los diferentes cursos.

### Ejercicio 1. Implementar desde el diagrama de clases

Dado que el secretario del director del colegio es experto en modelado de software, ha creado un diagrama de clases UML para indicarnos lo que se desea. Implemente las clases y los métodos Java que se describen en el diagrama de clases teniendo en cuenta los siguientes detalles. **No es necesario implementar otros métodos que los indicados en el diagrama:**

- En la clase Curso, la nota de aprobación del curso, el cupo de alumnos y el nombre del curso se reciben por parámetro en el constructor.
- En la clase Alumno, el método equals() verifica si todos los atributos del alumno son iguales a los atributos de otro alumno recibido por parámetro.



## Resolución:

Primero codificamos la derivación estructural del diagrama de clases:

### Clase Curso

```
public class Curso {  
  
    private List<Inscripcion> inscripciones;  
    private float notaAprobacion;  
    private String nombreCurso;  
    private int cupo;  
  
    public Curso (float nota, int cupo, String nombre){  
        this.notaAprobacion=nota;  
        this.cupo=cupo;  
        this.nombreCurso = nombre;  
        this.inscripciones = new ArrayList<Inscripcion>();  
    }  
  
    public String getNombreCurso() {  
        return nombreCurso;  
    }  
  
    public void setNombreCurso(String nombreCurso) {  
        this.nombreCurso = nombreCurso;  
    }  
    public float getNotaAprobacion() {  
        return this.notaAprobacion;  
    }  
}
```

### Clase Alumno

```
public class Alumno {  
  
    private int dni;  
    private String nombre;  
    private String apellido;  
    private int edad;  
  
    public Alumno(int dni, String nombre, String apellido, int edad){  
        this.dni=dni;  
        this.nombre=nombre;  
        this.apellido=apellido;  
        this.edad=edad;  
    }  
}
```

```

    public boolean equals(Object o){
        if (o == null || !(o instanceof Alumno))
            return false;

        Alumno a = (Alumno) o;
        if((this.dni==a.getDni()) && (this.nombre.equals(a.getNombre()) &&
            (this.apellido.equals(a.getApellido()) &&
            ((this.edad==a.getEdad()))
            return true
        else return false;
    }

}

```

### Clase Inscripcion

```

public class Inscripcion {

    private Curso curso;
    private Alumno alumno;
    private float nota;

    public Inscripcion (){}

    public void setCurso(Curso curso) {
        this.curso = curso;
    }

    public void setAlumno(Alumno alumno) {
        this.alumno = alumno;
    }

    public float getNota() {
        return nota;
    }

    public void setNota(float nota) {
        this.nota = nota;
    }

}

```

### Clase Colegio

```

public class Colegio{
    private List<Curso> cursos;

    public Colegio(){
        this.cursos=new ArrayList<Curso>();
    }
}

```

```
}
```

## Ejercicio 2. Implementar un método a partir de un enunciado

El director del colegio nos informó que es fundamental llevar las estadísticas del Programa 111 Mil en el colegio. Por eso, es importante que el director pueda:

- a) Dado un nombre de un curso por parámetro, obtener la cantidad de estudiantes en ese curso
- b) Dado un nombre de un curso por parámetro, obtener la cantidad de estudiantes que aprobaron el examen con la nota de aprobación o más.

Por favor, ayudá a los desarrolladores a implementar estos métodos de la clase `Colegio` con ayuda del diagrama de clases del Ejercicio 1. Escribí el código Java de los métodos que necesita el director.

### Resolución:

En la clase **Colegio**, se implementan los siguientes métodos:

```
public int obtenerCantAlumnos (String nombreCurso){

    int cantidadAlumnos=0;

    for(Curso curso:cursos)
    {
        String nombre = curso.getNombreCurso();
        if (nombre.equals(nombreCurso))
            cantidadAlumnos=curso.getInscripciones().size();
    }
    return cantidadAlumnos;
}

public int obtenerCantidadAprobados (String nombreCurso){
    int cantidadAprobados=0;

    boolean cursoEncontrado = false;
    for(int i=0; i<=cursos.size() && !cursoEncontrado; i++)
    {
        Curso curso= cursos.get(i);
        String nombre = curso.getNombreCurso();
        if (nombre.equals(nombreCurso)){
            Inscripcion inscripciones = curso.getInscripciones();
            for (Inscripcion insc : inscripciones)
                if(insc.getNota()>=curso.getNotaAprobacion())

                cantidadAprobados+=inscripcion.cantAprobados();
        }
    }

    return cantidadAprobados;
}
```

```
}
```

### Ejercicio 3. Interpretación de código

Un grupo de desarrolladores que trabajan con director del colegio nos pidió ayuda para interpretar el siguiente fragmento de código que pertenece a la clase Colegio:

```
public float xxx(Curso c){
    float yyy=0;
    for (Iterator<Curso> itCurso = cursos.iterator(); itCurso.hasNext();){
        Curso curso = itCurso.next();
        if(curso.getNombreCurso().equals(c.getNombreCurso())){
            List<Inscripcion> inscriptos= curso.getInscripciones();
            for (Iterator<Inscripcion> itInsc = inscriptos.iterator();
                itInsc.hasNext();)
            {
                Inscripcion inscripcion= itInsc.next();
                if(inscripcion.getNota() >= curso.getNotaAprobacion())
                    yyy++;
            }
        }
    }

    if (c.getInscripciones().size() > 0)
        return (yyy/c.getInscripciones().size())*100;
    else return 0;
}
```

- a) Por favor, ayudá a los desarrolladores a determinar cuál de las siguientes opciones se corresponde con lo que realmente hace el código.
- Calcula el promedio de alumnos aprobados en un curso dado.
  - Calcula el promedio de alumnos aprobados en todos los cursos.
  - Calcula el porcentaje de alumnos aprobados en un curso dado.
  - Calcula el porcentaje de alumnos aprobados en todos los cursos.
  - Calcula la cantidad de alumnos desaprobados en un curso dado.

La respuesta correcta es \_

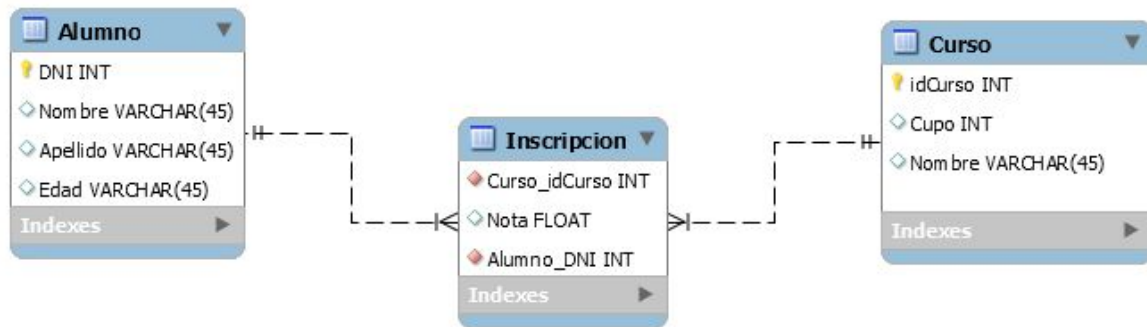
- b) ¿Qué nombre significativo le pondrías al método **xxx()**?
- c) ¿Qué nombre significativo le pondrías a la variable **yyy**?

#### Resolución:

- a) **Opción C.**
- b) Al método **xxx()** lo nombraría como **getPorcentajeAprobadosCurso()**.
- c) A la variable **yyy** la denominaría **cantidadAprobados**.

#### Ejercicio 4. Interpretación de DER.

El equipo de desarrolladores que trabajan con Adolfo, se encuentra con algunos problemas de interpretación sobre DER. Ayudá a los desarrolladores a decidir si las siguientes opciones son Verdaderas (V) ó Falsas (F) (Asignar a cada afirmación un valor V o F. Todas las afirmaciones tienen que tener un valor asignado. Puede haber varios V y varios F):



F	El atributo Nombre es la clave primaria de la entidad Alumno.
V	Existe una relación 1 a N entre Alumno e Inscripcion.
F	El atributo Cupo de la entidad Curso es clave foránea.
V	Los atributos Alumno_DNI y Curso_idCurso son claves foráneas de la entidad Inscripcion.
V	La relación entre las entidades Alumno y Curso es N a N.
F	Cupo es un atributo del tipo entero de la entidad Alumno
F	La relación entre Curso e Inscripcion es 1 a 1

#### Ejercicio 5. Escribir una consulta SQL.

El equipo administrativo que trabaja con el director del colegio se encuentra en este momento contabilizando los estudiantes que al menos aprobaron 1 curso de los que ofrece el Programa 111Mil en el Colegio Secundario N°1. En su base de datos existe información de los alumnos, de los cursos y de las inscripciones a dichos cursos. A partir del DER del ejercicio anterior, el equipo necesita listar el nombre y apellido de cada alumno, con su DNI, edad y el nombre del o los curso/s con nota 7 ó superior. El listado debe estar ordenado alfabéticamente por el apellido del alumno. El equipo necesita que escribas la consulta SQL correspondiente.

#### Resolución

```
SELECT Alumno.Nombre, Alumno.Apellido, Alumno.DNI, Alumno.Edad, Curso.Nombre FROM
Alumno LEFT JOIN Inscripcion ON Alumno.DNI = Inscripcion.Alumno_DNI
LEFT JOIN Curso ON Inscripcion.Curso_idCurso = Curso.idCurso
WHERE Inscripcion.Nota >= 7
ORDER BY Alumno.Apellido ASC;
```