

Contexto - Inscripción a Curso 111MIL

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.

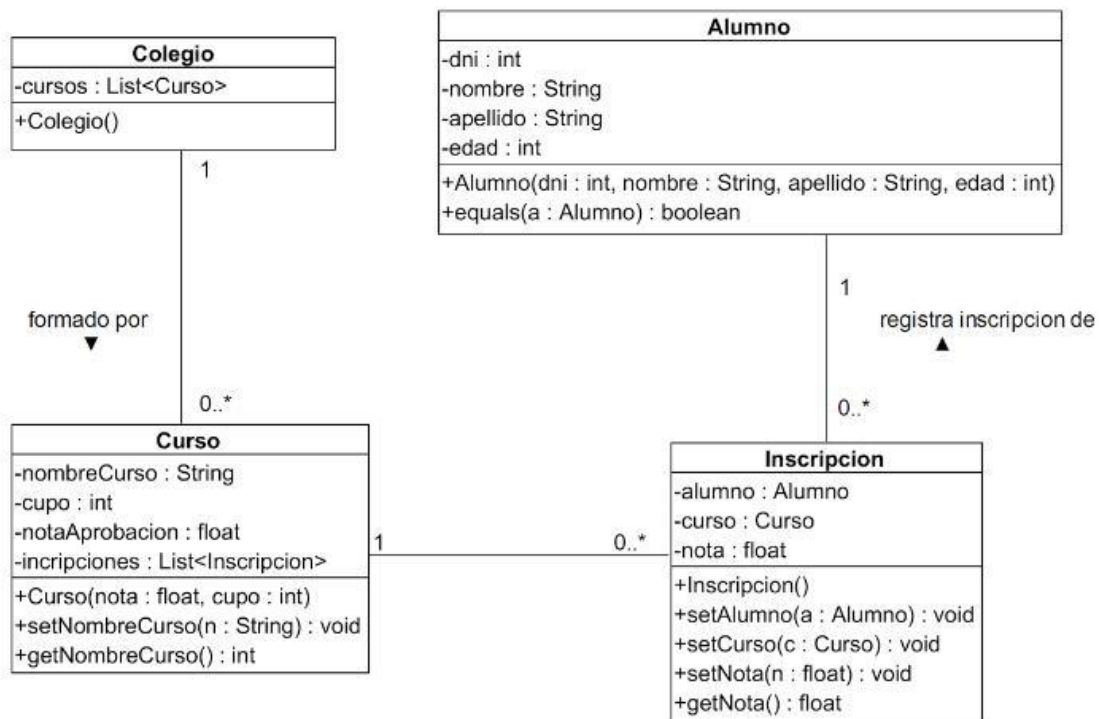
Ejercicio 1 - Implementar desde el diagrama de clases

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.

Implementá en los puntos a), b) y c) 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 y el cupo de alumnos 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.



1a- Completar:

Clase Curso

```
public class Curso {  
  
    private List<Inscripcion> inscripciones;  
  
    public Curso(float nota, int cupo) {  
        this.inscripciones = new ArrayList<Inscripcion>();  
    }  
  
    DEFINIR EL RESTO DE LAS VARIABLES Y MÉTODOS  
}
```

1b- Completar:

Clase Alumno

```
public class Alumno {  
  
    private int dni;  
  
    public Alumno(int dni, String nombre, String apellido, int edad) {  
        this.dni=dni;  
        this.nombre=nombre;  
        this.apellido=apellido;  
        this.edad=edad;  
    }  
  
    DEFINIR EL RESTO DE LAS VARIABLES Y MÉTODOS  
}
```

1c- Completar:

Clase Inscripcion

```
public class Inscripcion {  
  
    private Curso curso;  
  
    public Inscripcion(){}  
  
    public void setCurso(Curso curso) {  
        this.curso = curso;  
    }  
  
    DEFINIR EL RESTO DE LAS VARIABLES Y MÉTODOS  
}
```

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.

Ejercicio 3 - Interpretación de código

Un grupo de desarrolladores que trabajan con el 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> iterator = cursos.iterator(); iterator.hasNext();){  
        Curso curso = iterator.next();  
        if((curso.getNombreCurso().equals(c.getNombre()))){  
            List<Inscripcion> inscriptos= curso.getInscripciones();  
            for (Iterator<Inscripcion> iterator = inscriptos.iterator();  
                iterator.hasNext();)  
            {  
                Inscripcion inscripcion= iterator.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.

Marca solo un óvalo.

- ☐ a. Calcula el promedio de alumnos aprobados en un curso dado.
- ☐ b. Calcula el promedio de alumnos aprobados en todos los cursos.
- ☐ c. Calcula el porcentaje de alumnos aprobados en un curso dado.
- ☐ d. Calcula el porcentaje de alumnos aprobados en todos los cursos.
- ☐ e. Calcula la cantidad de alumnos desaprobados en un curso dado.

¿Qué nombre significativo le pondrías al método xxx()?

¿Qué nombre significativo le pondrías a la variable yyy?

Ejercicio 4 – Implementación de métodos a partir del siguiente enunciado

El director del colegio, ya nos había informado que es fundamental llevar las estadísticas del Programa 111 Mil en el colegio, pero también necesita de otras funcionalidades para realizar mejor su gestión. Por eso, a continuación, se detallan los nuevos requerimientos:

A- Dado un rango de edades por parámetro (edadMaxima y edadMinima), obtener la cantidad de estudiantes que estén dentro de ese rango de edades.

B- Dado un rango de edades por parámetro (edadMaxima y edadMinima), obtener el porcentaje de estudiantes que estén dentro de ese rango de edades.

C- Obtener un listado de aquellos apellidos que se repitan para dos o mas alumnos. Por ejemplo, si hay dos o mas alumnos con apellido Perez, entonces ese apellido debería aparecer en el listado. En cambio, si solo hay un alumno que tenga por apellido Cardozo, este apellido no debería aparecer en el listado.

D-

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

Ejercicio 5 – Implementación de métodos a partir del siguiente enunciado

Implementar en código Java los métodos necesarios para satisfacer los siguientes requerimientos:

A- El director debería poder seleccionar un curso a partir de un listado de los cursos que se dictan en el colegio, y a partir de esa selección, obtener el promedio de todas las notas que hayan alcanzado los alumnos. Se propone llamar al método con el nombre obtenerPromedioGeneral.

B- Implementar el método obtenerMayorNota(). Siguiendo la misma mecánica propuesta en el punto A.

C- Implementar el método obtenerMenorNota().

Por favor, ayudá a los desarrolladores a implementar estos métodos de la clase Curso con ayuda del diagrama de clases del Ejercicio 1.