

RETO 2.4

Nombre del reto:	Diseño de sistema estadístico para escuela																				
Descripción del reto con su respectiva solución:																					
Una escuela primaria desea implementar un sistema estadístico de clasificación de notas de los exámenes de sus estudiantes.																					
La escuela utiliza un sistema de notas basado en números que van desde el 0 hasta el 5, con diferentes rangos con labels que enfatizan el desempeño de los estudiantes.																					
El sistema debe generar datos estadísticos que puedan ayudar a los profesores a identificar grupos o estudiantes con dificultades. Actualmente se cuenta con la información de las calificaciones de los exámenes para cada uno de los estudiantes de la clase F, el sistema debe generar los datos estadísticos a partir de los siguiente información:																					
<table><tr><th>Nombre</th><th>Género</th><th>Materia</th><th>Nota</th></tr><tr><td>maria</td><td>f</td><td>matematicas</td><td>4.2</td></tr><tr><td>armando</td><td>m</td><td>matematicas</td><td>4.9</td></tr><tr><td>maria</td><td>f</td><td>idiomas</td><td>3.7</td></tr><tr><td>armando</td><td>m</td><td>idiomas</td><td>2.7</td></tr></table>		Nombre	Género	Materia	Nota	maria	f	matematicas	4.2	armando	m	matematicas	4.9	maria	f	idiomas	3.7	armando	m	idiomas	2.7
Nombre	Género	Materia	Nota																		
maria	f	matematicas	4.2																		
armando	m	matematicas	4.9																		
maria	f	idiomas	3.7																		
armando	m	idiomas	2.7																		
Las calificaciones en la escuela se asignan con la siguiente escala de rangos:																					
<table><tr><th>Rango de notas</th><th>Calificación</th></tr><tr><td>(4.5 - 5]</td><td>Excelente</td></tr><tr><td>(3.5 - 4.5]</td><td>Sobresaliente</td></tr><tr><td>(2.5 - 3.5]</td><td>Regular</td></tr><tr><td>(1 - 2.5]</td><td>Insuficiente</td></tr><tr><td>[0 - 1]</td><td>Deficiente</td></tr></table>		Rango de notas	Calificación	(4.5 - 5]	Excelente	(3.5 - 4.5]	Sobresaliente	(2.5 - 3.5]	Regular	(1 - 2.5]	Insuficiente	[0 - 1]	Deficiente								
Rango de notas	Calificación																				
(4.5 - 5]	Excelente																				
(3.5 - 4.5]	Sobresaliente																				
(2.5 - 3.5]	Regular																				
(1 - 2.5]	Insuficiente																				
[0 - 1]	Deficiente																				
El algoritmo debe ser capaz de responder a las siguientes preguntas:																					
<div><div>1.</div><div>¿Cuál es el porcentaje de aprobación para todos los exámenes presentados por el grupo?</div></div> <div><div>2.</div><div>¿Cuántos exámenes tienen una calificación insuficiente?</div></div> <div><div>3.</div><div>¿Cuál es la materia con el peor desempeño promedio para el género masculino?</div></div> <div><div>4.</div><div>¿Cuál es el estudiante con el mejor desempeño para la materia geografía?</div></div>																					
EJEMPLO																					
Para facilitar el proceso de ingreso y manipulación de los datos, los valores de tipo string se les asignará un identificador numérico único por categorías:																					

Nombre	Identificador
armando	1
nicolas	2
daniel	3
maria	4
marcela	5
alexandra	6

Materia	Identificador
geografia	1
matemáticas	2
informatica	3

Género	Identificador
m	0
f	1

Entrada del programa
18 1.0 0.0 1.0 1.6 1.0 0.0 2.0 3.5 1.0 0.0 3.0 1.3 2.0 0.0 1.0 0.3 2.0 0.0 2.0 4.4 2.0 0.0 3.0 2.0 3.0 0.0 1.0 4.2 3.0 0.0 2.0 3.9 3.0 0.0 3.0 3.2 4.0 1.0 1.0 4.3 4.0 1.0 2.0 2.7 4.0 1.0 3.0 3.9 5.0 1.0 1.0 1.8 5.0 1.0 2.0 1.8 5.0 1.0 3.0 0.2 6.0 1.0 1.0 1.6 6.0 1.0 2.0 0.4 6.0 1.0 3.0 2.4
Salida del programa
0.44 7 geografia maria

Otro aspecto importante es el formato de entrada de los datos, la primera línea de la entrada se trata del número de registros que se deben leer. Las líneas de los registros tienen un formato de tabla, el orden de las columnas es: nombre, género, materia y calificación, las columnas están separadas por un espacio. Se recomienda copiar y pegar este ejemplo en la terminal para realizar pruebas.

FASE 2

El sistema debe implementarse utilizando el paradigma de programación orientada a objetos(POO), los requerimientos funcionales del programa son los siguientes:

- Se debe implementar todo el procesamiento de información y cálculos en una clase llamada **SchoolGradingSystem**.
- La clase **SchoolGradingSystem** debe contener un método llamado **readData**, el cual no debe recibir ni retornar ningún parámetro. El objetivo de este método es leer los valores de entrada del programa.
- La clase debe contener 4 métodos, estos métodos no deben recibir ningún parámetro y la salida de estos métodos debe ser el valor que da respuesta a cada uno de las preguntas. Los métodos deben ser nombrados **question#**, donde **#** corresponde al número de la pregunta a la cual debe dar respuesta, para este caso serían los números del 1 al 4(se debe mantener el orden presentado en este documento).
- Además de la clase anterior se debe crear un método main en una clase llamada App, en esta se debe instanciar un objeto de tipo **SchoolGradingSystem**, posteriormente se deben realizar lo siguiente:
 - Leer los datos utilizando el método correspondiente.
 - Realizar los cálculos y operaciones necesarias, llamando a los métodos correspondientes.
 - Imprimir los resultados en el orden correspondiente y con el formato requerido.
- Utilizar las buenas prácticas de programación.

Notas:

- Es importante seleccionar los tipos adecuados para cada método, ya que esto puede generar errores a la hora de calificar.
- Prestar especial cuidado a las notaciones de los rangos.
- Se considera aprobado a una calificación igual o mayor a Regular.
- Los elementos con un identificador menor tienen prioridad, al ejecutar un proceso y este arroja varios posibles resultados, se debe imprimir el que tenga menor identificador.

Instrucciones Reto 1

Nota: Por favor **NO** incluya mensajes en los inputs.

Nota: Las tildes y cualquier otro signo ortográfico han sido omitidos a propósito en las entradas y salidas del programa. Por favor **NO** use ningún signo ortográfico dentro del desarrollo de su solución ya que estos pueden representar errores en la calificación automática de Codegrade.

Nota: Los archivos deben llamarse **SchoolGradingSystem.java** y **App.java**, de lo contrario no podrán ser cargados en la plataforma de Codegrade

Nota: En caso de haber salidas numéricas todas deben estar formateadas dependiendo del tipo de dato, los reales(float o double) deben ser impresos con una precisión de dos decimales, ejemplo: 2.67, 5.00 y 12.40. Los de tipo entero(int o long) deben ser impresos como enteros, ejemplo: 3, 45 y 10.