

RETO 5.4

| | |
|------------------|--|
| 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:

| 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:

| 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:

1. ¿Cuál es el porcentaje de aprobación para todos los exámenes presentados por el grupo?
2. ¿Cuántos exámenes tienen una calificación insuficiente?
3. ¿Cuál es la materia con el peor desempeño promedio para el género masculino?
4. ¿Cuál es el estudiante con el mejor desempeño para la materia geografía?

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 5

Para esta fase del proyecto se aprovechara lo anteriormente construido en la fase 3:

Sistema de calificación por GUI:

Anteriormente se solicitó la construcción de 2 clases **GradingSystem** y **SchoolGradingSystem**, ahora se va a extender la aplicación de programación orientada a objetos en el proyecto :

- Se debe crear una clase llamada **Student**, esta clase tiene como objetivo representar un objeto estudiante dentro del proyecto, las características de esta clase son las siguientes.
 - Esta clase debe contener la información correspondiente a un estudiante: Nombre, Género, lista de las calificaciones correspondientes a ese estudiante las cuales están asociadas a una asignatura en específico.
 - Se debe hacer uso del constructor de la clase para recibir los parámetros de la información del estudiante (los mencionados anteriormente).
- Las clases anteriormente construidas **GradingSystem** y **SchoolGradingSystem** deben hacer uso de la clase **Student**, esto significa que los métodos deben hacer uso de los objetos tipo Student y no de la estructura de datos que anteriormente se había usado para almacenar los datos.
- La interfaz gráfica puede ser reutilizada y actualizada para hacer uso de los nuevos cambios en la estructura de clases.
- Se debe realizar un diagrama de clases UML del proyecto.
- Para este proyecto no se requiere el uso de una base de datos.

Sistema estadístico

18

CALCULAR

1.0 0.0 1.0 4.9
1.0 0.0 2.0 2.0
1.0 0.0 3.0 4.6
2.0 0.0 1.0 2.2
2.0 0.0 2.0 5.6
2.0 0.0 3.0 6.3
3.0 0.0 1.0 3.5
3.0 0.0 2.0 2.0
3.0 0.0 3.0 0.9
4.0 1.0 1.0 6.2
4.0 1.0 2.0 9.7
4.0 1.0 3.0 5.1
5.0 1.0 1.0 8.7
5.0 1.0 2.0 5.9
5.0 1.0 3.0 7.8
6.0 1.0 1.0 6.3

0.33
1
geografia
marcela

Notas:

- Los datos presentes en la imagen de ejemplo son de demostración y no muestran resultados referentes a este enunciado en específico.
- 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 5

Nota: En la actividad Reto 5 GUI deberá incluir un archivo .zip con el contenido de la carpeta src del proyecto Java. Por favor llame el comprimido de la siguiente forma: **ApellidoNombre-#Grupo.zip**. Donde Apellido se refiere al apellido de estudiante, Nombre se refiere al nombre del estudiante y #Grupo se refiere al número de grupo al que pertenece el estudiante.

Nota: En la actividad Reto 5 Codegrade deberá incluir los archivos SchoolGradingSystem, GradingSystem y Student con el nombre que se especifica en el enunciado del reto.