**RETO 3.31**

|  |  |
| --- | --- |
| 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** | | marcela | f | matemáticas | 4.2 | | armando | m | matemáticas | 4.5 | | marcela | f | idiomas | 3.8 | | 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. ¿Cuántos de los exámenes tienen una calificación mayor a la del promedio del grupo? 2. ¿Cuántos exámenes tienen una calificación Regular? 3. ¿Cuál es la materia con el mayor numero de examenes reprobados 4. ¿Cuál es el estudiante con el mejor desempeño para la materia química?     **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 | | quimica | 1 | | idiomas | 2 | | historia | 3 |  |  |  | | --- | --- | | Género | Identificador | | m | 0 | | f | 1 |  |  | | --- | | Entrada del programa | | 18 1.0 0.0 1.0 1.1 1.0 0.0 2.0 1.8 1.0 0.0 3.0 3.3 2.0 0.0 1.0 4.3 2.0 0.0 2.0 1.6 2.0 0.0 3.0 4.3 3.0 0.0 1.0 3.8 3.0 0.0 2.0 0.7 3.0 0.0 3.0 1.8 4.0 1.0 1.0 0.6 4.0 1.0 2.0 2.5 4.0 1.0 3.0 4.0 5.0 1.0 1.0 0.5 5.0 1.0 2.0 3.8 5.0 1.0 3.0 1.2 6.0 1.0 1.0 2.6 6.0 1.0 2.0 4.6 6.0 1.0 3.0 0.1 | | Salida del programa | | 9  2  idiomas  nicolas |     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 3**  Para el diseño del programa usted deberá:   * Implementar POO creando una super clase llamada **Persona**. * Implementar POO creando una subclase llamada **Estudiante**. * Implementar un método dentro de clase **Estudiante** llamado **contarRegulares** que devuelva un entero con la cantidad de exámenes que tienen una calificación Regular (no recibir datos por parámetro) * Implementar una clase llamada reto3 en donde se encuentre el método principal de ejecución del programa, y en la cual se instancien los objetos de tipo **Estudiante**. * Utilizar las buenas prácticas de programación.   Adicionalmente deberá diseñar una interfaz de usuario por medio de la cual se ingresen los datos al programa cuyos requerimientos mínimos son:   * Utilizar campos de texto para ingressar los datos. * Utilizar labels para identificar correctamente las entradas o cualquier información que se desee brindar. * Utilizar botones para procesar los datos ingresados.   Ejemplo de interfaz de usuario  En la interfaz anterior el botón Ingresar incluye los datos en el campo de datos de entrada, y el botón procesar calcula la salida deseada y la muestra en el campo de datos de salida.  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. * No es necesario replicar con exactitud la interfaz de usuario mostrada de ejemplo, usted puede presentar un diseño propio que cumpla con los requerimientos minimos solicitados para la GUI. | |