

**Fundamentos de Programación Proyecto final de la asignatura 2023-I**

**Samuel Alejandro Monsalve Sarmiento**

**Carlos Mario Niño Ariza**

**Fundación Universitaria Konrad Lorenz**

**Facultad de matematicas e ingenierias, Fundamentos de programación**

**Bogotá D.C, 24 de mayo de 2023**

**Problema:** Normalmente, los estudiantes universitarios o de bachillerato presentan la dificultad de organizar sus deberes y asignaturas por la sobrecarga académica que presentan, por lo cual, no saben administrar adecuadamente su tiempo para realizar cada una de sus actividades. Así mismo, los alumnos no conocen a detalle su proceso académico, pues cada nota que reciben se sube al sistema del instituto al que pertenecen, pese a que los docentes dan a conocer el resultado de cada estudiante, estos no toman nota del desarrollo de sus calificaciones. Por ende, es pertinente un programa que ayude a organizar su deberes y tareas y un sistema que les ayude a subir sus notas para así mismo promediarlas, lo que les ayudará a ser más autónomos y organizados.

**Título:** Tasklist and grade point average

**Objetivo:** El objetivo del programa es la creación de un programa para ayudar a los estudiantes a organizar sus actividades y subir sus notas para promediarlas para que conozcan su proceso académico

**Descripción:** Crear un programa que reciba las tareas pendientes del usuario junto con su fecha límite, Además, le solicitará las notas de cada asignatura para promediarlas.

### **Ejecución del código**

Al ejecutar el código, en primera instancia solicita al usuario que ingrese la cantidad de materias que está cursando, luego en base al número de asignaturas, pide el nombre de cada una de las materias ingresadas por el teclado, sucesivamente la cantidad de tareas pendientes por cada asignatura. En función de ello, solicita el nombre del deber pendiente, la fecha de entrega, dado el caso que sea más de una tarea, solicitará que ingrese el nombre de cada una y su correspondiente fecha límite de entrega. Acto seguido, el programa pide el número de notas que

lleva en esa materia, de acuerdo a la cantidad ingresadas, realiza el promedio de calificaciones para saber la definitiva de la asignatura. Realizará el mismo proceso con la cantidad de ingresada con cada una de las variables anteriormente mencionadas.

### **Pasos y explicación del código**

1. Se crea un objeto “Scanner” para leer la entrada del usuario
2. Mediante `System.out.println`, se le solicita al usuario que ingrese la cantidad de materias que está cursando
3. Se crea una nueva variable llamada “cantidadMaterias”, mediante `int cantidadMaterias = sc.nextInt()`, el usuario ingresara el número de asignaturas
4. Se usa `“sc.nextLine()”` para limpiar el búfer de entrada después de leer el número de materias para evitar que se capture una línea vacía en la siguiente lectura
5. Se crean los siguientes arreglo: “materias”, “tareas”, “fechasEntrega”, “promediosNotas”  
.
6. `“String[] materias = new String[cantidadMaterias];”` y `double[] promediosNotas = new double[cantidadMaterias];` Estas líneas declaran un arreglo unidimensional.
7. `“String[][] tareas = new String[cantidadMaterias][];”` y `“String[][] fechasEntrega = new String[cantidadMaterias][];”`. Aquí se declara un arreglo bidimensional
8. Comienza un bucle “for” que itera desde 0 hasta “cantidadMaterias - 1” para obtener la correspondiente información de cada materia.
9. Dentro del bucle se solicita al usuario que ingrese el nombre de la materia y se almacena en el arreglo “materias”.
10. Se solicita al usuario que ingrese la cantidad de tareas pendientes para esa materia

11. El programa lee el número ingresado por el usuario y se almacena en la variable “cantidadTareas”
12. Se utiliza “sc.nextLine()” para limpiar el búfer de entrada después de leer la cantidad de tareas.
13. Se crean los arreglos “tareas[i]” y “fechasEntrega[i]” con el tamaño correspondiente a la variable “cantidadTareas”
14. Se inicia un bucle “for” anidado al anterior para obtener la información de cada tarea.
15. Dentro de ese bucle “for”, se solicita al usuario que ingrese el nombre de la tarea y se almacena en “tareas[i][j]”.
16. Luego, se solicita al usuario que ingrese la fecha de entrega de la tarea y se almacena en el arreglo “fechasEntrega[i][j]”
17. Después de completar la información de las tareas, se solicita al usuario que ingrese la cantidad de notas para esa materia.
18. Se lee el número ingresado por el usuario y se almacena en la variable “cantidadNotas”
19. Se utiliza “sc.nextLine()” para limpiar el búfer de entrada después de leer la cantidad de notas.
20. Se inicia otro bucle “for” para obtener las notas y calcular el promedio.
21. Dentro de este bucle, se solicita al usuario que ingrese cada nota y se suma a la variable “sumaNotas”.
22. Una vez que se han obtenido todas las notas, calcula el promedio dividiendo “sumaNotas” entre “cantidadNotas” y se almacena en el arreglo “promediosNotas[i]”.

23. Si no es la última materia (es decir “i” no es igual a “cantidadMaterias - 1”), se utiliza “`sc.nextLine()`” para limpiar el búfer de entrada y evitar problemas de lectura en la siguiente iteración del bucle.
24. Después de completar la información de todas las materias, se muestra un resumen de las tareas pendientes y su fecha de entrega y los promedios de las notas.
25. Se itera a través de las materias y para cada una se muestran las tareas pendientes y sus fechas de entrega.
26. Finalmente, se muestra el promedio de notas para cada materia.