

1. Objetivos

Ejercitar conceptos básicos de programación C++, implementando un programa y su correspondiente documentación que resuelva el problema descrito más abajo.

2. Alcance

Este trabajo práctico es de elaboración grupal, evaluación individual, y de carácter obligatorio para todos alumnos del curso.

3. Requisitos

El trabajo deberá ser entregado personalmente, en la fecha estipulada, con una carátula que contenga los datos completos de todos los integrantes, un informe impreso de acuerdo con lo que mencionaremos en la sección 6, y con una copia digital de los archivos fuente necesarios para compilar el trabajo.

4. Descripción

En este trabajo práctico se implementará un programa para poder obtener datos específicos sobre un conjunto de mediciones de distintos sensores. Los datos a obtener serán el promedio, máximo y mínimo medido dentro de un rango, junto con la cantidad de mediciones usadas para obtenerlos. Las consultas (o *queries*) pueden buscar información de un sensor en específico o del promedio de todos estos.

Para comprender esto mejor, se puede ver con un ejemplo. Supongamos que tenemos los datos de los sensores de temperatura de una PC, en particular, los datos del sensor del CPU, del motherboard y del GPU. Se requiere el dato en un cierto punto en el tiempo del CPU, pero, para reducir el error, se decide obtener el de un cierto rango. Entonces, se obtendrá el promedio de estos valores, y el máximo y mínimo para dar una idea de qué tanto error hay en la medición. A su vez, puede buscarse el dato en el mismo instante pero en promedio de todos los sensores, para dar una idea de la temperatura que hay dentro del gabinete, por lo que se obtendrían los mismos datos (promedio, máximo y mínimo) pero los promedios totales se a su vez se promediarán entre si.

Para simplificar el problema, en vez de leer los *timestamps* se entenderá que los sensores miden a intervalos regulares, donde el primer dato será el momento 0 y se avanzará de a uno desde ese. Existe la posibilidad que no se haya hecho la medición de un sensor en un momento, lo cual no contará para ningún tipo de promedio, mínimo o máximo.

4.1. Formato de entrada y salida

Los archivos que contengan la base de datos serán archivos de valores separados por comas (o CSV), donde en la primera línea contendrán los nombres de los sensores y luego las líneas siguientes serán cada medición, empezando por el momento en el que se hizo esta y luego tendrá los datos de cada sensor en el orden especificado por la primera línea. Los archivos de consultas también serán CSV, donde cada línea representa una consulta que especificará el sensor y el rango (primer y último instante donde buscará los datos). De querer usar la información de todos los sensores, no especificará el sensor. La Fig. 1 muestra un ejemplo de esto para dos sensores A y B: ID_A e ID_B son los IDs de los sensores y los a_i , b_i son las mediciones tomadas por estos a intervalos regulares.

La salida tendrá, por cada consulta, el promedio, el mínimo, el máximo y la cantidad de datos, en ese orden. De no haber ningún dato en ese rango, escribirá "NO DATA".

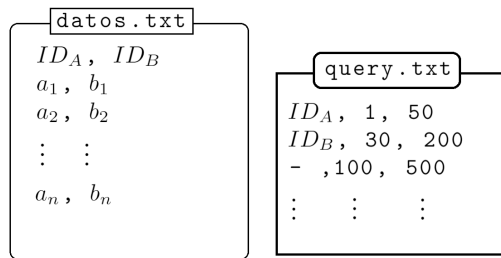


Figura 1: Formato de los archivos. El guión significa que no hay ningún dato

4.2. Interfaz

La interacción con el programa es a través de la línea de comando. Se incluirán las siguientes opciones:

- `-d`, o `--data`, que incluirá el archivo de donde se leerá la base de datos. Esta opción es obligatoria.
- `-i`, o `--input`, que incluirá el archivo de donde se leerán las consultas. Si su argumento es `"-"` (sin las comillas) o si no está especificada, se leerá de la entrada standard, `std::cin`.
- `-o`, o `--output`, que incluirá el archivo donde se escribirá la salida. Como antes, si su argumento es `"-"` o si no está especificada, se leerá de la salida standard, `std::cout`.

4.3. Ejemplos

El ejemplo más simple consiste en una entrada vacía. Observar que la salida es también vacía:

```
$ touch entrada1.txt
$ ./tp0 -d data.txt -i entrada1.txt -o salida1.txt
$ cat salida1.txt
$
```

Ahora, consideremos el caso que funciona correctamente con una consulta:

```
$ cat data2.txt
sensor1, sensor2
1, 1
2, 2
6, 6
$ cat entrada2.txt
sensor1, 0, 3
$ ./tp0 -d data2.txt -i entrada2.txt
3.0, 1, 6, 3
```

Para los casos de que no haya ningún dato en el rango, no se encuentre el sensor, o la consulta esté con un formato incorrecto, tenemos:

```
$ cat entrada3.txt
sensor2, 8, 12
sensor1, 0, 3
$ ./tp0 -d data2.txt -i entrada3.txt
NO DATA
3.0, 1, 6, 3

$ cat entrada4.txt
sensor4, 0, 3
sensor1, 0, 3
$ ./tp0 -d data2.txt -i entrada4.txt
UNKNOWN ID
3.0, 1, 6, 3
```

```
$ cat entrada5.txt
sensor1, A, 3
sensor1, 0, 3
$ ./tp0 -d data2.txt -i entrada5.txt
BAD QUERY
3.0, 1, 6, 3
```

Se incluirá un archivo de ejemplos.

5. Optativos

En el trabajo se pueden agregar distintos componentes.

Se puede considerar que se puede obtener los datos de un subconjunto de sensores, en cuyo caso la consulta, dirá, por ejemplo, "*sensor1;sensor3;sensor4, 5, 40*", indicando que se busca el promedio de los datos de los sensores 1,3 y 4, en el intervalo [5,40).

También se puede considerar que los datos de los sensores están separados en distintos archivos, que contienen la información de uno o más sensores cada uno, pero que no aparece el mismo sensor en dos archivos diferentes (si el sensor1 y el sensor2 están en el archivo1, no están en el archivo2). En este caso, probablemente sea mejor utilizar una bandera en la línea de comandos para que el programa funcione sabiendo esto. De usarse, esta bandera será -m, o -multiple.

6. Informe

El informe deberá incluir:

- Documentación relevante al diseño e implementación del programa, incluyendo una descripción a alto nivel del código.
- Documentación relevante a los algoritmos involucrados en la solución del trabajo.
- Documentación relevante al proceso de compilación: cómo obtener el ejecutable a partir de los archivos fuente.
- Las corridas de prueba, con los comentarios pertinentes.
- El código fuente, en lenguaje C++ (en dos formatos, digital e impreso).
- Este enunciado.

7. Fechas

La última fecha de entrega es el 27 de septiembre.