

TICS311: Tarea #3

Universidad Adolfo Ibañez

Felipe Aguilera
felipe.aguilera@edu.uai.cl

Danilo Bórquez Paredes
danilo.borquez.p@uai.cl

Sebastián Sáez
sebastian.saez@edu.uai.cl

03 de Mayo de 2021

Objetivos

- Internalizar conceptos de tablas de hash y listas
- Trabajo con memoria dinámica y estructuras
- Uso de archivos de cabecera

1. El curso ahora es ONLINE


El mismo profesor del trabajo anterior nuevamente complicado (parece que su vida es así...)! Luego de su excelente trabajo obteniendo los promedios de sus alumnos, y ordenando, ahora necesita manejar un curso online con **MUCHOS** estudiantes! Por eso se le ha pedido a usted como experto en estructura de datos que genere un programa que permita manejar un gran volumen de alumnos de manera eficiente.

Para cumplir este objetivo se le pide que guarde cada alumno en una estructura compuesta por el nombre (**string**) y el promedio de notas (**float**). Estas estructuras serán acomodadas en **listas**, que a su vez estarán en una tabla Hash donde cada “balde” corresponderá a la inicial del apellido del alumno, como muestra el ejemplo de la figura 1.

El programa debe leer el archivo con los datos de entrada, guardarlos en la estructura definida por la Figura 1 y mostrar por pantalla los nombres de los 10 mejores promedios (promedios más altos). Además, debe mostrar todos los alumnos cuyo apellido comience con una letra ingresada por el usuario, ordenados de manera descendente según su promedio.

1.1. Entrada del programa

Su programa recibirá como entrada la ruta del archivo que contiene todas las notas de los alumnos de un curso, tal como se muestra en el ejemplo siguiente.



```
TICS311@localhost:~$ ./tarea3 notas.csv
```

El nombre del archivo (notas.csv) es sólo un ejemplo, dado que puede llamarse de cualquier manera. El formato de este archivo sí es estricto, y a continuación se muestra un ejemplo:

notas.csv							
nombre	prueba 1	prueba 2	prueba 3	prueba 4	proyecto	tareas	
Danilo Borquez Paredes	1.8	1.9	1.1	5.5	4.1	4.5	
Felipe Aguilera	2.9	4.4	4.1	6.9	5.8	4.4	
Sebastian Saez	1	2.5	5.9	4.7	3.5	1.4	

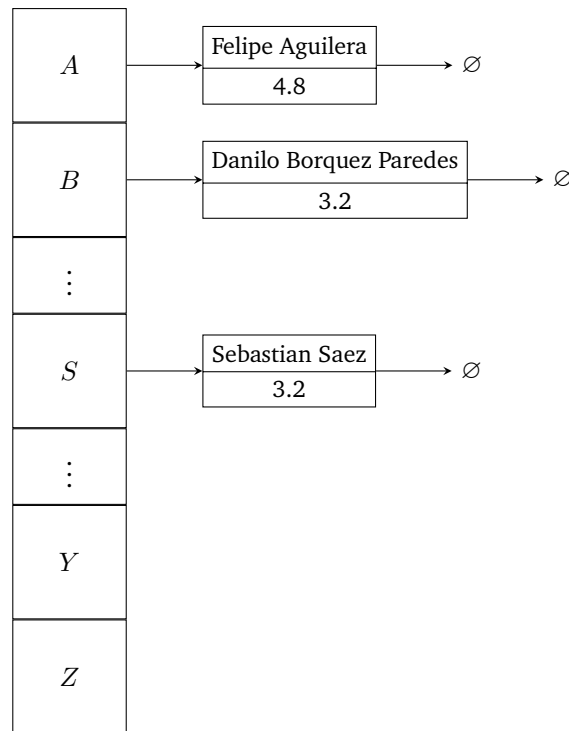


Figura 1: Estructura de datos de los alumnos

Las características del archivo son las siguientes:

- La cantidad de alumnos es variable, pero siempre hay **a lo menos 1**.
- La cantidad de notas es variable, pero siempre hay **a lo menos 1**.
- La primera línea del archivo siempre contiene la información de cada instrumento de evaluación.
- Cada dato siempre estará separado por una coma (,).
- El separador de decimales será el punto (.).
- Nunca habrán campos vacíos (no habrán dos comas seguidas).
- Los instrumentos de evaluación y los nombres de los alumnos pueden tener espacios.
- No habrán caracteres especiales en los nombres como tildes o ñ.

1.2. Salida del programa

El programa debe mostrar por la terminal los 10 alumnos con mayor promedio en orden decreciente según sus promedios aritméticos. Además debe pedir al usuario una letra (puede ser mayúscula o minúscula), y debe mostrar todos los alumnos cuyo apellido comience por esa letra, ordenados en forma descendente según su promedio aritmético.

```

TICS311@localhost:~$ Felipe Aguilera: 4.8
Sebastian Saez: 3.2
Danilo Borquez Paredes: 3.2

Ingrese una letra: B
Danilo Borquez Paredes: 3.2

```

2. Sobre la entrega

- La tarea debe ser hecha en lenguaje de Programación C.
- Puede utilizar cualquier algoritmo de ordenamiento.
- Se evaluará orden y modularidad de su código.
- Cada grupo puede ser de 2 o 3 personas.
- La tarea se debe entregar el día **Viernes 28 de Mayo** a las 23:59.
- Por cada día de atraso se descontará 1 punto, comenzando a las 00:00 horas del siguiente día. Por ejemplo si entrega la tarea a las 00:00 del siguiente día, la nota máxima que puede obtener es un 6.0
- Para la corrección se utilizará un compilador gcc v 5.1 o superior
- La entrega se realiza por la plataforma Webcursos¹
- El archivo a entregar debe ser un zip que contenga una carpeta en su interior (y sólo una carpeta) con el nombre **tarea3**. Dentro de esa carpeta debe haber un Makefile² y por lo menos un main.c. Además el nombre de su zip debe ser grupoX-tarea3.zip, donde X es el número de su grupo.
- **IMPORTANTE:** el directorio **tarea3** debe contener el Makefile que generará el archivo ejecutable **tarea3**.

¹<http://webcursos.uai.cl>

²Este archivo deben generarlo. Información útil pueden encontrarla en <https://stackoverflow.com/questions/1484817/how-do-i-make-a-simple-makefile-for-gcc-on-linux>. Pueden probar su Makefile en un computador con MAC o Linux, o en VSCode en la terminal con UBUNTU.