

# TICS311: Tarea #4

## Universidad Adolfo Ibáñez

Felipe Aguilera  
felipe.aguilera@edu.uai.cl

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

Sebastián Sáez  
sebastian.saez@edu.uai.cl

1 de Junio de 2021

## Objetivos

- Internalizar conceptos de árboles
- Uso de archivos de cabecera

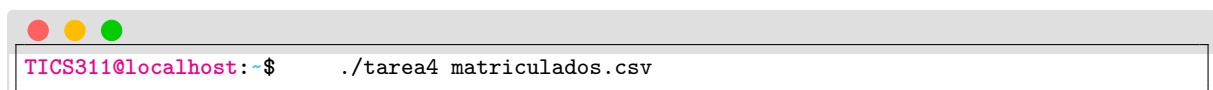
## 1. Matrículas abiertas

Al parecer las clases están funcionando, porque ahora se han abierto matrículas para diferentes cursos. Luego de su excelente trabajo con las listas y tablas de hash, ahora se necesita manejar un proceso de matrícula para muchos alumnos. 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 matrículas de manera eficiente.

Para cumplir este objetivo se le pide que guarde cada alumno en una estructura compuesta por el nombre del alumno (**string**) y la nota de presentación al curso (**float**). Estas estructuras serán acomodadas en un **árbol AVL**. El programa debe leer el archivo con los datos de entrada y guardarlos en el árbol, según las notas de presentación. Finalmente debe generar un archivo “resumen” llamado **out.txt**, el cual tiene en su interior la cantidad de alumnos, el promedio de nota de presentación, la nota mínima y la nota máxima.

### 1.1. Entrada del programa

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



```
TICS311@localhost:~$ ./tarea4 matriculados.csv
```

El nombre del archivo (matriculados.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:

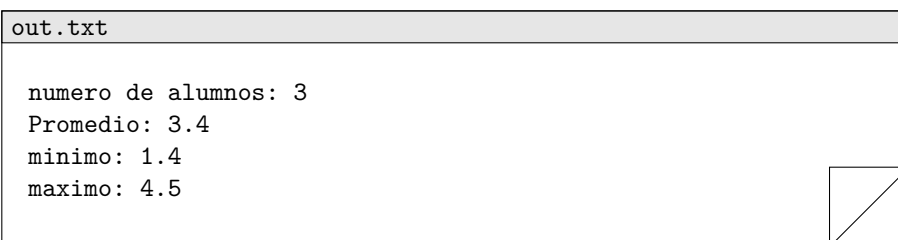
matriculados.csv	
nombre,Nota de presentacion	
Danilo Borquez Paredes,4.5	
Felipe Aguilera,4.4	
Sebastian Saez,1.4	

Las características del archivo son las siguientes:

- La cantidad de alumnos 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 generar un archivo **out.txt** que contiene un resumen de la matrícula.



```
out.txt

numero de alumnos: 3
Promedio: 3.4
minimo: 1.4
maximo: 4.5
```

## 2. Sobre la entrega

- La tarea debe ser hecha en lenguaje de Programación C.
- Debe obtener el mínimo y el máximo directamente de su árbol AVL.
- 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 18 de Junio** 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 Webcursos<sup>1</sup>
- El archivo a entregar debe ser un zip que contenga una carpeta en su interior (y sólo una carpeta) con el nombre **tarea4**. Dentro de esa carpeta debe haber un Makefile<sup>2</sup> y por lo menos un main.c. Además el nombre de su zip debe ser grupoX-tarea4.zip, donde X es el número de su grupo.
- **IMPORTANTE:** el directorio **tarea4** debe contener el Makefile que generará el archivo ejecutable **tarea4**.

<sup>1</sup><http://webcursos.uai.cl>

<sup>2</sup>Este 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.