Trabajo Práctico Especial Programación Imperativa Febrero 2021

1. Objetivo

Diseñar e implementar un programa para el <u>procesamiento de datos de árboles en el espacio público de una ciudad</u>, basado en datos reales. Para ello se deberá realizar tanto el *front-end* como el *back-end*, este último <u>basado en la creación de al menos un TAD</u>.

Para este trabajo se busca poder procesar los datos de los árboles de dos ciudades: Ciudad Autónoma de Buenos Aires, Argentina y Vancouver, British Columbia, Canadá . Ambos datos son extraídos de los respectivos portales de gobierno en formato CSV.

2. Descripción funcional

El programa consiste en dos ejecutables distintos, uno para cada dataset.

Cada uno de estos ejecutables deberá leer dos archivos CSV: uno con la información de los árboles correspondientes a una ciudad y otro archivo con la información de los barrios de esa ciudad. En ambos el delimitador es un punto y coma (";").

A continuación se listan los dos ejemplos de uso que se busca para la aplicación: procesar los datos de árboles de la Ciudad Autónoma de Buenos Aires z y de Vancouver . Sin embargo, es importante recordar que la mayor parte de la implementación no debe estar atada a la realidad de los ejemplos de uso. Por ejemplo, los barrios serán los que la aplicación obtenga en ejecución a partir del archivo de barrios y no serán aceptadas implementaciones que tengan fijos estos datos. En otras palabras, la implementación deberá funcionar también para procesar los datos de árboles de cualquier otra ciudad, manteniendo siempre la estructura de los archivos que se presenta a continuación. Si bien se detalla el origen de los archivos, los mismos fueron modificados para simplificar su procesamiento.

Los archivos tienen las siguientes características:

Datos de árboles de Buenos Aires , a partir de ahora arbolesBUE.csv

♦ Origen:

https://data.buenosaires.gob.ar/dataset/arbolado-publico-lineal/archivo/ecf38a47-563f-42c1-9bd 4-7cedf35d536b

- **Descarga:** En Campus ITBA.
- **Cantidad de líneas:** 370.089 (incluyendo el encabezado)
- **Campos Relevantes:**
 - **comuna:** Nombre del barrio donde se encuentra el árbol.
 - calle nombre: Nombre de la calle donde se encuentra el árbol
 - nombre_cientifico: Nombre científico del árbol
 - diametro altura pecho: Diámetro del árbol a la altura del pecho

El archivo se compone de una primera línea de encabezado, con los títulos de cada campo. De la segunda línea en adelante, cada línea representa un árbol conteniendo los datos de cada uno de los campos, separados por ";".

nro_registro;tipo_activ;comuna;manzana;calle_nombre;calle_altura;di
reccion_normalizada;nombre_cientifico;estado_plantera;ubicacion_plantera;
nivel plantera;diametro altura pecho;altura arbol

79838;Lineal;1; ;Eyle Petrona;0;EYLE, PETRONA 47;Platanus x acerifolia;Ocupada;Fuera de línea;A nivel;1;17

. . .

Datos de barrios de Buenos Aires , a partir de ahora barriosBUE.csv

- ❖ Descarga: En Campus ITBA. El archivo es un subconjunto del dataset, conteniendo únicamente los campos relevantes.
- **Cantidad de líneas:** 16 (incluyendo el encabezado)
- **Campos:**
 - **nombre:** Nombre del barrio, para relacionarlo con el campo **comuna** de arbolesBUE.csv.
 - habitantes: Cantidad de habitantes del barrio.

El archivo se compone de una primera línea de encabezado. De la segunda línea en adelante, cada línea representa un barrio.

nombre; habitantes

2;149607

14;227003

15;182427

. . .

Datos de árboles de Vancouver , a partir de ahora arbolesVAN.csv

- Origen: https://opendata.vancouver.ca/explore/dataset/street-trees/
- **Descarga:** En Campus ITBA.
- **♦ Cantidad de líneas:** 146.746 (incluyendo el encabezado)
- **Campos Relevantes:**
 - **NEIGHBOURHOOD NAME:** Nombre del barrio donde se encuentra el árbol.
 - STD_STREET: Nombre de la calle donde se encuentra el árbol
 - COMMON NAME: Nombre científico del árbol
 - **DIAMETER**: Diámetro del árbol a la altura del pecho

El archivo se compone de una primera línea de encabezado, con los títulos de cada campo. De la segunda línea en adelante, cada línea representa un árbol conteniendo los datos de cada uno de los campos, separados por ";".

TREE_ID; CIVIC_NUMBER; STD_STREET; GENUS_NAME; SPECIES_NAME; CULTIVAR_NA ME; COMMON_NAME; ASSIGNED; ROOT_BARRIER; PLANT_AREA; ON_STREET_BLOCK; ON_STREET; NEIGHBOURHOOD_NAME; STREET_SIDE_NAME; HEIGHT_RANGE_ID; DIAMETER; CURB; DATE_P LANTED; Geom

20666;2838; W 19TH AV; PRUNUS; CERASIFERA; ATROPURPUREUM; PISSARD PLUM; N; N; 10; 2800; W 19TH AV; ARBUTUS-RIDGE; EVEN; 3; 20.0; Y;; "{""type"": ""Point"", ""coordinates"": [-123.16858, 49.25546]}"

Datos de barrios de Vancouver , a partir de ahora barrios VAN.csv

- ❖ Descarga: En Campus ITBA. El archivo es un subconjunto del dataset, conteniendo únicamente los campos relevantes.
- **❖** Cantidad de líneas: 23 (incluyendo el encabezado)
- ***** Campos:
 - nombre: Nombre del barrio, para relacionarlo con el campo NEIGHBOURHOOD NAME de arbolesVAN.csv.
 - habitantes: Cantidad de habitantes del barrio.

El archivo se compone de una primera línea de encabezado. De la segunda línea en adelante, cada línea representa un barrio.

nombre; habitantes

WEST POINT GREY;13065 HASTINGS-SUNRISE;34575 KERRISDALE;13975

. . .

Se asume que el formato y contenido de los archivos es correcto.

El programa deberá recibir por línea de comando (no por entrada estándar) el path del archivo de árboles y el path del archivo de barrios.

El ejecutable para procesar los datos de debe llamarse arbolesADTBUE. El ejecutable para procesar los datos de debe llamarse arbolesADTVAN.

Por ejemplo, si se desea procesar los archivos CSV de y se llaman arboles.csv y barrios.csv y están en el mismo directorio que el ejecutable arbolesADTBUE, el programa se puede invocar como:

\$> ./arbolesADTBUE arboles.csv barrios.csv

Si ambos archivos CSV se llaman arb.csv y bar.csv y están en el directorio superior al ejecutable arbolesADTBUE, se invocará como

\$> ./arbolesADTBUE ../arb.csv ../bar.csv

Una vez recibido correctamente los path de los archivos CSV, el programa deberá resolver las consultas (queries) que se listan a continuación, dejando los resultados de cada una en archivos .csv con el nombre pedido y localizados en el mismo directorio que el ejecutable.

Resolver todas las consultas con una única lectura de cada uno de los archivos y de una sola vez. No se aceptarán implementaciones que ofrezcan un menú de opciones o similar que permita al usuario decidir cuál de las consultas procesar.

Los ejemplos de salida que se listan a continuación tienen valores ficticios y no coinciden con la salida esperada para los archivos CSV.

En caso de rendir en la <u>primera fecha</u> de final podrán formar grupos de <u>tres o</u> <u>menos alumnos</u> e implementar <u>las dos primeras</u> *queries*. En caso de formar grupos de <u>cuatro alumnos</u> deberán implementar <u>las tres primeras</u> *queries*.

En caso de rendir en la <u>segunda fecha</u> de final podrán formar grupos de <u>tres o</u> <u>menos alumnos</u> e implementar <u>las cuatro primeras queries</u>. En caso de formar grupos de <u>cuatro alumnos</u> deberán implementar <u>las cinco primeras queries</u>.

En caso de rendir en la <u>tercera fecha</u> de final deberán formar grupos de <u>tres o</u> <u>menos alumnos</u> e implementar <u>todas</u> las *queries*. <u>No se aceptarán grupos de cuatro alumnos</u> en la tercera fecha.

Query 1: Diámetro promedio por barrio

Donde cada línea de la salida contenga, separados por ";" el nombre del barrio y el promedio del diámetro del árbol a la altura del pecho de los árboles pertenecientes a ese barrio.

El orden de impresión es descendente por diámetro y luego alfabético por nombre de barrio.

El promedio del diámetro del árbol a la altura del pecho debe imprimirse <u>truncado</u> a dos decimales.

Sólo se deben listar los barrios presentes en el archivo CSV de barrios.

Sólo se deben considerar los árboles que tengan un diámetro positivo.

- □ Nombre del archivo: query1.csv
- □ Salida de ejemplo para □:

BARRIO; DIAMETRO PROMEDIO

15;37.36

14;37.21

5;34.88

13;34.57

11;34.47

. . .

☐ Salida de ejemplo para [1]:

BARRIO; DIAMETRO_PROMEDIO KITSILANO; 14.64 SHAUGHNESSY; 14.13 DUNBAR-SOUTHLANDS; 13.93 WEST POINT GREY; 13.23 KERRISDALE; 12.02

Query 2: Para cada barrio, la especie con mayor cantidad de árboles por habitante

Donde cada línea de la salida contenga, separados por ";" el nombre del barrio, el nombre de la especie con mayor cantidad de árboles por habitante de ese barrio y el valor numérico correspondiente. La cantidad de árboles por habitante de un barrio consiste en el cociente entre el total de árboles localizados en ese barrio y el número de habitantes del mismo. Por lo tanto, para un barrio y una especie, el valor numérico a considerar consiste en el cociente entre la cantidad de árboles de esa especie ubicados en ese barrio y la cantidad de habitantes de ese barrio.

Sólo se deben listar los barrios presentes en el archivo CSV de barrios.

El valor numérico debe imprimirse truncado a dos decimales.

El orden de impresión es alfabético por nombre de barrio.

- □ Nombre del archivo: query2.csv
- □ Salida de ejemplo para □:

BARRIO; NOMBRE CIENTIFICO; ARBOLES POR HABITANTE

1;Fraxinus pennsylvanica;0.01 10;Fraxinus pennsylvanica;0.06 11;Platanus x acerifolia;0.02 12;Fraxinus pennsylvanica;0.06 13;Fraxinus pennsylvanica;0.03

☐ Salida de ejemplo para [1]:

BARRIO; NOMBRE_CIENTIFICO; ARBOLES_POR_HABITANTE

ARBUTUS-RIDGE; PISSARD PLUM; 0.04

DOWNTOWN; RED MAPLE; 0.00

DUNBAR-SOUTHLANDS; KWANZAN FLOWERING CHERRY; 0.03

FAIRVIEW; RED MAPLE; 0.00

GRANDVIEW-WOODLAND; KWANZAN FLOWERING CHERRY; 0.02

Query 3: Calle con más árboles por barrio

Donde cada línea de la salida contenga, separados por ";" el nombre del barrio, el nombre de la calle con más árboles de ese barrio y la cantidad de árboles que hay en esa calle de ese barrio.

Sólo se deben listar los barrios presentes en el archivo CSV de barrios.

El orden de impresión es alfabético por el nombre del barrio.

- ☐ Nombre del archivo: query3.csv
- ☐ Salida de ejemplo para ☐:

BARRIO; CALLE CON MAS ARBOLES; ARBOLES

1; Saenz Peña Roque, Pres. Diagonal Norte Av.; 123 10; Lascano; 436 11; Lamarca Emilio; 712 12;Quesada;523 13;3 de Febrero;439

□ Salida de ejemplo para [1]:

BARRIO; CALLE CON MAS ARBOLES; ARBOLES

ARBUTUS-RIDGE; W KING EDWARD AV; 276 DOWNTOWN; PACIFIC BOULEVARD; 173 DUNBAR-SOUTHLANDS; W KING EDWARD AV; 463 FAIRVIEW; W BROADWAY; 399 GRANDVIEW-WOODLAND; VICTORIA DRIVE; 295

Query 4: Mínimo y máximo diámetro por especie de árbol

Donde cada línea de la salida contenga, separados por ";" el nombre de la especie del árbol, el mínimo diámetro del árbol a la altura del pecho de esa especie y el máximo diámetro del árbol a la altura del pecho de esa especie.

El orden de impresión es alfabético por especie.

Todos los valores deben imprimirse con dos decimales.

- ☐ Nombre del archivo: query4.csv
- ☐ Salida de ejemplo para ☐:

NOMBRE_CIENTIFICO; MIN; MAX

Abies alba; 5.00; 156.00 Abutilon grandiflorum; 0.00;77.00 Abutilon pictum; 3.00; 3.00 Acacia baileyana; 0.00; 65.00 Acacia caven; 0.00; 169.00

. . .

☐ Salida de ejemplo para [1]:

NOMBRE_CIENTIFICO; MIN; MAX ACCOLADE CHERRY; 3.00; 42.00 AKEBONO FLOWERING CHERRY; 0.00; 66.00 ALDER SPECIES; 0.00; 26.50 ALDERLEAFED MOUNTAIN ASH; 2.00; 19.75

ALIA'S MAGNOLIA;5.00;5.50

. . .

Query 5: Especies de árboles que están presentes en todos los barrios

Donde cada línea de la salida contenga el nombre de la especie del árbol que tiene 1 o más árboles en todos los barrios presentes en el archivo CSV de barrios.

Sólo se deben listar los barrios presentes en el archivo CSV de barrios.

El orden de impresión es alfabético por especie.

- ☐ Nombre del archivo: query5.csv
- ☐ Salida de ejemplo para ☐:

NOMBRE_CIENTIFICO

Acacia visco
Acer buergerianum
Acer negundo
...
Tipuana tipu

□ Salida de ejemplo para [1]:

Ulmus procera

NOMBRE_CIENTIFICO

AKEBONO FLOWERING CHERRY AMERICAN SWEETGUM ARISTOCRAT PEAR

• • •

WESTERN RED CEDAR WILLOW OAK

Ouery 6: Top 10 barrios con mayor diversidad de especies

Donde cada línea de la salida contenga, separados por ";" el nombre del barrio y la cantidad de especies distintas que hay en ese barrio.

Sólo se deben listar los barrios presentes en el archivo CSV de barrios.

Sólo se deben listar los 10 barrios con mayor cantidad de especies distintas.

El orden de impresión es descendente por cantidad de especies distintas y luego alfabético por nombre de barrio.

- ☐ Nombre del archivo: query6.csv
- ☐ Salida de ejemplo para ☐:

BARRIO; ESPECIES_DISTINTAS

9;268

8;247

15;236

12;234

11;230

4;230

10;229

7;221

13;214

6;185

☐ Salida de ejemplo para [1]:

BARRIO; ESPECIES_DISTINTAS

DUNBAR-SOUTHLANDS; 364

RENFREW-COLLINGWOOD; 352

HASTINGS-SUNRISE;351

KITSILANO; 327

KENSINGTON-CEDAR COTTAGE; 325

RILEY PARK; 312

GRANDVIEW-WOODLAND; 311

SUNSET;311

KERRISDALE; 289

SHAUGHNESSY; 288

Para todas las queries, tener en cuenta que todos los archivos de salida deben contener la línea de encabezado correspondiente indicada en las salidas de ejemplo.

Las salidas de ejemplo de cada query presentadas en esta consigna fueron obtenidas con subconjuntos del dataset provisto por lo que <u>no deben comparar el resultado obtenido con los valores expresados en las salidas de ejemplo</u>. Estos resultados son sólo a modo explicativo para que queden ejemplificados los criterios de ordenación y formato esperado de los resultados.

3. Contenidos

Para la realización del presente trabajo se recomienda consultar el siguiente material:

- Para leer argumentos por línea de comandos: https://www.tutorialspoint.com/cprogramming/c_command_line_arguments.htm
- strtok puede resultar útil para "parsear" las líneas de los archivos: https://www.tutorialspoint.com/c_standard_library/c_function_strtok.htm
- **Presentación de Archivos:** Contiene lo esencial para el manejo de archivos en C. Tener en cuenta que se procesarán únicamente archivos de texto en forma secuencial, por lo que las funciones a utilizar serán fopen, fgets, fclose, fprintf o similares. Se encuentra en Campus ITBA.

De todas formas, los alumnos pueden consultar dudas relacionadas a estos nuevos contenidos usando los foros de discusión de Campus ITBA. No se responderán consultas que no sean formuladas a través del Campus.

4. Diseño e Implementación del Programa

Se debe realizar un diseño donde se separe claramente la lectura y preparación de datos (*front-end*) del almacenamiento y procesamiento de los datos (*back-end*).

Recordar que ninguna función de back-end debe invocar a una función de front-end.

Para el almacenamiento de los datos se deberá desarrollar al menos un TAD.

En ningún caso se debe repetir código para resolver situaciones similares, sino que debe implementarse una correcta modularización y se deben reutilizar funciones parametrizadas.

Tanto la biblioteca como el *front-end* deben estar correctamente comentados y en el caso de la biblioteca se debe escribir el archivo de encabezado correspondiente.

El programa no debe presentar leaks de memoria.

El programa no debe abortar por ningún motivo y ante cualquier error se debe mostrar un mensaje adecuado.

Se tendrá en cuenta la eficiencia en el uso de recursos, tanto el tiempo de ejecución como la memoria utilizada.

5. Uso de Git

Es obligatorio el uso de un repositorio Git para la resolución de este final. Siguiendo los pasos indicados en la clase de Taller de Git deberán crear un repositorio en GitHub donde todos los integrantes del grupo colaboren con las modificaciones del código provisto. No se aceptarán entregas que utilicen un repositorio git con un único commit que consista en la totalidad del código a entregar.

Los distintos *commits* deben permitir ver la evolución del trabajo, tanto grupal como **individual**

Muy importante: los repositorios creados deben ser privados.

6. Material a entregar

Cada grupo deberá subir al Campus ITBA un archivo compactado conteniendo como mínimo los siguientes archivos:

- Archivos fuentes y de encabezado.
- Archivo de texto README con la explicación de cómo generar los ejecutables y de cómo ejecutarlos.
- makefile. Deberá generar los dos ejecutables pedidos. Consultar apunte en Campus ITBA.

Revisar que en el archivo comprimido también se encuentra el directorio oculto .git/donde se detallan todas las modificaciones realizadas.

7. Armado de Grupos

Los alumnos deben informar la conformación del grupo y la fecha de final elegida en el Foro de Discusión "TPE Final" de Campus ITBA al menos un día antes de la fecha de entrega correspondiente.

Cada grupo deberá realizar la entrega mediante la actividad correspondiente en Campus ITBA:

- En caso de rendir en la <u>primera fecha</u> de final (04/02/2021 11:00 ART) deberán entregar antes del 02/02/2021 23:59 ART.
- En caso de rendir en la <u>segunda fecha</u> de final (11/02/2021 11:00 ART) deberán entregar antes del 09/02/2021 23:59 ART.
- En caso de rendir en la tercera fecha de final (18/02/2021 11:00 ART) deberán entregar antes del 16/02/2021 23:59 ART.

En todos los casos deben enviar sólo el material pedido. No incluir código compilado, archivos de prueba publicados, etc.

En todos los casos el grupo completo deberá asistir a la Sala Virtual de Campus ITBA en un día y hora previamente acordado (se asignarán turnos para todos los grupos que rindan el final ese día). En la Sala Virtual se comunicará la nota del final y los integrantes del grupo rendirán un coloquio, el cual podrá modificar la nota individual de los integrantes del grupo.

Para que el trabajo sea aceptado todos los integrantes del grupo deben estar inscriptos en fecha de final que acordaron en el armado del grupo.

En caso de inscribirse y no entregar en fecha, se calificará como ausente. Si algún miembro del equipo no se presenta en el horario previamente acordado el alumno será calificado como ausente, y deberá conformar otro grupo en otra fecha de final, no afectando la evaluación de los alumnos que se presenten.

Todos los integrantes de cada grupo <u>deberán inscribirse</u> por su cuenta en la fecha de final elegida en el SGA.

8. Criterios de Evaluación y Calificación

El programa debe poder compilarse y ejecutarse <u>en Pampero</u> usando **gcc** con los parámetros **-pedantic -std=c99 -Wall -fsanitize=address**. El último flag verifica que la memoria no sea usada en forma incorrecta (extenderse de la cantidad de bytes reservados en una zona, acceder a una zona de memoria liberada, no liberar memoria antes de finalizar la ejecución, etc.).

Recomendamos utilizar todos los flags desde el principio del desarrollo para encontrar y resolver antes los posibles errores que surjan. Es útil agregar además el flag de *debug* -g para obtener más información sobre los errores encontrados por el -fsanitize.

No se aceptará el uso de bibliotecas de terceros, a excepción de la biblioteca estándar de C. El código debe ser íntegramente de autoría propia. El uso de bibliotecas no autorizadas implicará la desaprobación.

Si un trabajo presenta errores de compilación, el mismo será reprobado. Se espera que no se presenten "warnings" evitables.

Para la evaluación y calificación del trabajo especial se considerarán:

- o el cumplimiento del modo de ejecución y de los nombres de archivos pedidos
- o el correcto funcionamiento del programa
- o la modularización realizada
- o el correcto diseño del TAD, que debería poder ser fácilmente adaptable para otras codificaciones, nombres o formatos de archivos
- o la separación del front-end y el back-end
- o la claridad del código
- o el cumplimiento de las reglas de estilo de programación dadas en clase
- o la eficiencia del procesamiento de los datos y el uso de la memoria.

9. Dudas sobre el TPE

Si bien el enunciado contempla la funcionalidad completa a desarrollar es normal que surjan dudas acerca de cómo interpretar ciertos casos. O que una consigna genere más de una posible solución, por lo que es importante que analicen bien el enunciado, y ante cualquier duda pregunten. Sólo se contestarán dudas sobre el enunciado. Las mismas deben volcarse al Foro de Discusión "TPE Final" del Campus ITBA.