

## Objetivo

El objetivo de este proyecto es poner en práctica los conceptos aprendidos en clase acerca de las estructuras de datos lineales: Lista, Pila y Cola, así como sobre algoritmos de ordenamiento y búsquedas eficientes de información.

## Contexto

“La accidentalidad vial en Colombia se ha convertido en la segunda manera de muerte violenta y según la Organización Mundial de la Salud (OMS), en el mundo cerca de seis millones de personas mueren por este evento, convirtiéndose así el tráfico automotor en la primera manera de muerte violenta en el nivel mundial.” Medicina Legal, Lesiones no Intencionales.

<http://www.medicinalegal.gov.co/documents/20143/49484/Muertes+Transito.pdf/ad2ae841-ed99-d524-66bf-1b70dec0b44a>

Para prevenir los accidentes viales los gobiernos usan policías de tránsito, cámaras, etc. los cuales producen mucha información. Para que las políticas del gobierno sean más efectivas esta información debe ser tomada en cuenta en la toma de decisiones. Sin embargo el problema es que la información no siempre es accesible al encargado de tomar las decisiones.

En este proyecto se va a procesar la información de infracciones viales de Washington D.C. para producir información en un formato más apropiado para tomar decisiones.

## Las Fuentes de Datos

A continuación, se presenta una descripción de las fuentes de datos que se utilizarán en el proyecto. Los datos relacionados se pueden descargar de la página Web: <http://opendata.dc.gov/datasets?q=moving%20violations>.

Específicamente los datos para cada mes del año 2018 están en:

- <http://opendata.dc.gov/datasets/moving-violations-issued-in-january-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-february-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-march-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-april-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-may-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-june-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-july-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-august-2018>

- <http://opendata.dc.gov/datasets/moving-violations-issued-in-september-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-october-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-november-2018>
- <http://opendata.dc.gov/datasets/moving-violations-issued-in-december-2018>

Los datos se pueden descargar de cada URL en formato CSV (*Comma separated values*) (sección Datos >> opción Descargar >> Hoja de cálculo). Un archivo CSV es un archivo de texto donde cada línea tiene sus campos de información separados por coma (,). A continuación se muestra un ejemplo de uno de los archivos:

```
OBJECTID,ROW_,LOCATION,ADDRESS_ID,STREETSEGID,XCOORD,YCOORD,TICKETTYPE,FINEAMT,TOTALPAID,PENALTY1,PENALTY2,ACCIDENTINDICATOR,AGENCYID,TICKETISSUEDATE,VIOLATIONCODE,VIOLATIONDESC,ROW_ID
15304300,,DC295      SW      .7      MILES      S/O      EXIT      1
S/B,810381,6993,398406.31,127737.13,Moving,100,0,0,,No,25,2018-12-
01T08:25:00.000Z,T119,SPEED 11-15 MPH OVER THE SPEED LIMIT,
15304301,,NEW      YORK      AVE      W/B      @      NEW      JERSEY
NW,808095,6936,398718.3825,137500.5723,Moving,150,0,0,,No,25,2018-12-
01T21:24:00.000Z,T113,FAIL TO STOP PER REGULATIONS FACING RED SIGNAL,
15304302,,NEW      YORK      AVE      W/B      @      NEW      JERSEY
NW,808095,6936,398718.3825,137500.5723,Moving,150,0,0,,No,25,2018-12-
02T11:15:00.000Z,T113,FAIL TO STOP PER REGULATIONS FACING RED SIGNAL,
15304303,,BLOCK      100      NEW      YORK      AVE      NW
WESTBOUND,814461,12462,398891.13,137577.22,Moving,30,0,0,,No,2,2018-12-
27T00:00:00.000Z,T333,FAIL TO DISPLAY PROOF OF VEHICLE INSURANCE,
```

Los atributos son:

- **OBJECTID:** Identificador único de la infracción.
- **ROW\_:**
- **LOCATION:** Dirección en formato de texto.
- **ADDRESS\_ID:** ID de la dirección.
- **STREETSEGID:** ID del segmento de la calle.
- **XCOORD:** Coordenada X donde ocurrió (No corresponde a longitud geográfica).
- **YCOORD:** Coordenada Y donde ocurrió (No corresponde a latitud geográfica).
- **TICKETTYPE:**
- **FINEAMT:** Cantidad a pagar por la infracción USD.
- **TOTALPAID:** Cuanto dinero efectivamente pagó el que recibió la infracción en USD.
- **PENALTY1:** Dinero extra que debe pagar el conductor.
- **PENALTY2:** Dinero extra que debe pagar el conductor.
- **ACCIDENTINDICATOR:** Si hubo un accidente o no.
- **AGENCYID:**
- **TICKETISSUEDATE:** Fecha cuando se puso la infracción.
- **VIOLATIONCODE:** código de la infracción.
- **VIOLATIONDESC:** descripción textual de la infracción.
- **ROW\_ID:**

## Carga de Información

Para responder a los requerimientos presentados más adelante, usted deberá cargar la información de todos los archivos .CSV correspondientes a uno de los cuatrimestres del año 2018. El cuatrimestre lo podrá seleccionar el usuario. Solo es permitido leer una vez la información de los archivos. Al final de la carga hay que reportar el número de infracciones de cada mes cargado y el total de infracciones del cuatrimestre.

## Impresión en consola de muchos datos

En algunos de los requerimientos el conjunto de datos retornado puede ser muy grande, por lo tanto se imprimirán solo los primeros N datos, donde N es una constante que inicialmente se puede igualar a 20.

## Requerimientos - Parte A (estudiante 1 de cada grupo)

**1A- Verificar que OBJECTID es en realidad un identificador único.** Verificar si todos los OBJECTID son únicos. Mostrar un texto indicando si No hay ningún OBJECTID repetido entre todos los meses. Si hay OBJECTID repetidos, mostrar aquellos que se encuentren repetidos.

Ord por ObjectID : nlog n

**2A- Consultar infracciones por fecha/hora inicial y fecha/hora final.** Para las infracciones resultantes mostrar OBJECTID y TICKETISSUEDAT. La Vista debe recibir una cola con las infracciones.

Ordenar iteracion y comp : n

**3A- Dado un tipo de infracción (VIOLATIONCODE) informar el (FINEAMT) promedio cuando no hubo accidente y el (FINEAMT) promedio cuando si lo hubo.**

if, sumar

**4A- Consultar las infracciones en una dirección (ADDRESS\_ID) en el rango fecha inicial y fecha final. Ordenar descendentemente por STREETSEGID y fecha.** Para las infracciones resultantes mostrar OBJECTID, TICKETISSUEDAT, STREETSEGID y ADDRESS\_ID. La Vista debe recibir una pila con las infracciones.

ordenar con comp creado

## Parte B (estudiante 2 de cada grupo)

**1B- Consultar los tipos de infracciones (VIOLATIONCODE) con su valor (FINEAMT) promedio en un rango dado.** Por cada tipo mostrar su VIOLATIONCODE y el FINEAMT promedio. La Vista debe recibir una cola con los tipos de infracciones y su respectivo FINEAMT promedio.

if: violC : n+1  
vc  
fineamt

**2B- Consultar infracciones donde la cantidad pagada (TOTALPAID) esta en un rango dado. Se ordena por fecha de infracción.** Para las infracciones resultantes mostrar OBJECTID, TICKETISSUEDAT, TOTALPAID. Se debe poder seleccionar si el resultado se

extract TP > ... ; orden  
comp fun(input)

retorna descendientemente o ascendientemente por fecha de la infracción. La Vista debe recibir una **pila** con las infracciones.

**3B- Consultar infracciones por hora inicial y hora final, ordenada ascendientemente** por VIOLATIONDESC. Para las infracciones resultantes mostrar OBJECTID, TICKETISSUEDAT y VIOLATIONDESC. La Vista debe recibir una **cola** con las infracciones.

**4B- Dado un tipo de infracción (VIOLATIONCODE) informar el (FINEAMT) promedio y su desviación estándar.**

*La it & extract; ordenar*  
*if,  $n-1$  ; calc*

## Parte C (trabajo en grupo)

**1C-El número de infracciones que ocurrieron en un rango de horas del día. Se define el rango de horas por valores enteros en el rango [0, 24]. Considerar todas las infracciones en el cuatrimestre.**

**2C-Grafica ASCII con el porcentaje de infracciones que tuvieron accidentes por hora del día. Tomar el 100% como todas las infracciones** que ocurrieron en todo el cuatrimestre. Se espera una gráfica de barras de este estilo:

Porcentaje de infracciones que tuvieron accidentes por hora. 2018

Hora	% de accidentes
00	X
01	X
02	XX
03	XXXXX
04	XXXXXXXXX
05	XXXXXXXXXX
06	XXXXXXXXXX
07	XXXXXXXXXX
08	XXXXXXXXXXXX
09	XXXXXXXXXXXXXX
10	XXXXXXXXXXXXXX
11	XXXXXXXXXXXXXX
12	XXXXXXXXXXXXXX
13	XXXX
14	XXXXXX
15	XXXXXXXXXXXXXXXX
16	XXXXXXXXXXXX
17	XXXXXX
18	XXXXXXXXXXXXXXXX
19	XXXXXXXXXXXX
20	XXX
21	XXXXX
22	XXXX
23	XX

Cada X representa Y%

Son libres de escoger el porcentaje que representa cada "X" pero en la grafica deben indicarlo.

3C- **La deuda** (TOTALPAID – FINEAMT - PENALTY1 – PENALTY2) total por infracciones que se dieron en un **rango** de fechas.

4C- **Grafica ASCII con la deuda acumulada total por infracciones.** La grafica es mes a mes comenzando con el primer mes del cuatrimestre. En el mes 1 se muestra la deuda total por infracciones en el mes 1; en el mes 2 se muestra la deuda total por infracciones en el mes 1 y el 2; en el mes 3 se muestra la deuda total por infracciones en el mes 1, 2 y 3; y así sucesivamente.

Se espera una grafica de este estilo:

```
Deuda acumulada por mes de infracciones. 2018
Mes | Dinero
01  | X
02  | XX
03  | XXXXXX
04  | XXXXXXXXXX
```

Cada X representa \$YYYY USD

Son libres de escoger el valor que representa cada “X” pero en la grafica deben indicarlo.

## Restricciones

- Los datos contenidos en los archivos sólo se pueden leer una vez
- Se deberá trabajar en Java 8
- El proyecto se debe implementar en Eclipse
- La entrada/salida de información adicionales se debe realizar por consola
- **No usar las colecciones del API Java.**

## Entrega de Diseño (33% Nota del Proyecto)

- Fecha/Hora límite de entrega: 17 Febrero, 11:59 p.m.
- Repositorio Bitbucket/GitHub con el nombre de la forma Proyecto\_1\_201910\_sec\_Y\_team\_Z (reemplazar Y por el número de la sección de su curso y Z su número de grupo)
- Verificar que el repositorio tiene configurado como usuarios a los monitores y al profesor con acceso de lectura.
- Entregables:
  - Documento con:
    - Documentación de los requerimientos funcionales (incluyendo los datos de entrada, la descripción, los datos de salida y estimación de complejidad temporal de cada requerimiento funcional)
    - Diseño de las Estructuras de Datos a utilizar (diagrama de clases UML, imagen)
    - Diseño de la Solución al proyecto (diagrama de clases UML, imagen)
  - Proyecto Eclipse/Java con:
    - Implementación y Pruebas Unitarias Automáticas de las Estructuras de Datos (Eclipse/Java)

- Lectura/carga de los datos para un cuatrimestre dado por el usuario (mostrar la evidencia que los datos fueron leídos/cargados)
- Hacer el desarrollo del proyecto en la rama (*branch*) master del repositorio. Como parte final de esta entrega, crear la rama (*branch*) entrega-diseNo. Verificar que en esta rama queda la copia de los entregables del proyecto. Los entregables en la rama entrega-diseNo No deben actualizarse después de la fecha/hora límite de esta entrega.

## Entrega Final (67% Nota del Proyecto)

- Fecha/Hora límite de entrega: 3 Marzo, 11:59 p.m.
- Repositorio Bitbucket/GitHub Proyecto\_1\_201910\_sec\_Y\_team\_Z (reemplazar Y por el número de la sección de su curso y Z su número de grupo) creado para la entrega de diseño. El proyecto completo (documentación e implementación) debe estar accesible en la rama (*branch*) master.
- Entregables:
  - Proyecto Eclipse/Java con:
    - Implementación completa de los requerimientos funcionales
- Los entregables en la rama master No pueden tener fecha de actualización posterior a la fecha/hora límite de esta entrega