

Cátedra de Algoritmos y Estructuras de Datos

Trabajo Práctico 2

Gestión de Cabinas de Peajes (v2.0)

La empresa *Peajes de Sudamérica*, conociendo el éxito de la versión 1.0 desarrollada para el TP1, solicita ahora una versión 2.0 del programa que permita obtener estadísticas a partir de todas las transacciones del último mes.

Otra vez, el programa deberá procesar datos de vehículos cuyas patentes pueden ser de cualquiera de los países del Mercosur, de acuerdo al mismo modelo anterior, *pero ahora se agrega también la patente chilena*, cuyo formato es ligeramente diferente según las gráficas siguientes:

Patentes Mercosur	
País miembro	Ejemplo
Argentina	AG 759 LH
Brasil	LSN4149
Bolivia	BB 01234
Paraguay	AAAA 123
Uruguay	ABC 1234

Patentes Chile	
BB·BB·10 CHILE	

Igual que en la versión 1.0 (TP1), los patrones de las patentes Mercosur son los siguientes:

- Argentina: LLNNNLL
- Brasil: LLLNLNN
- Bolivia: LLNNNNN
- Paraguay: LLLLNNN
- Uruguay: LLLNNNN

En esos patrones una **L** representa *letras*, y una **N** representa *números o dígitos*. Todas las patentes Mercosur tienen la misma longitud: 7(siete) caracteres en total (los espacios en blanco que ocasionalmente pueden aparecer en la figura adjunta o en una placa real, se colocan solo a los efectos de facilitar la lectura por parte de los humanos, pero NO se almacenan (ni deben ser cargados) internamente al gestionar una de estas cadenas en una variable).

Pero las placas chilenas tienen 6(seis) caracteres, y el formato es LLLLNN: cuatro letras, seguidas de dos dígitos, como se ve en la gráfica anterior. Otra vez, los espacios en blanco que aparecen en la figura o en una placa real, aparecen para facilitar la lectura por parte de las personas, pero no forman parte de la patente en sí misma.

Igual que en la versión 1.0, sólo a los efectos de este trabajo (*y sin que esto sea necesariamente real*), se asume que todos los países Mercosur cobran un **importe base** por cada peaje equivalente a 300 pesos, salvo Brasil que asumiremos que cobra peajes en una **base** de 400 pesos, y Bolivia que cobra peajes en una **base** de 200 pesos. Para simplificar, asuma que en todos los países ese monto está expresado en pesos argentinos (y no se preocupe por este detalle).

El **importe básico** que cada vehículo pagó debe calcularse ahora nuevamente, considerando lo mismo que en la versión 1.0: si el vehículo es una motocicleta se aplica un descuento del 50% sobre el **importe base** que

cobra esa cabina, y si el vehículo es un camión se aplica un recargo del 60% sobre el **importe base** de la cabina. Solo los automóviles pagan un importe básico igual al importe base. Distinga por favor entre el **importe base (400, 300 o 200)** y el **importe básico (igual al base si es auto, la mitad del base si es moto, o el base más el 60% si es camión)**.

Del mismo modo que en la versión 1.0, debe calcularse también otra vez el **importe final del ticket** considerando que si la forma de pago fue por telepeaje se aplica un descuento del 10% al **importe básico** indicado en el párrafo anterior. Y si el **pago fue manual**, entonces **el importe final es igual al importe básico**.

Para esta versión 2.0 el programa deberá procesar los datos de muchos vehículos (y no solo de uno, como en la versión 1.0). Los datos de todos esos vehículos no serán cargados por teclado, sino que vendrán almacenados en un *archivo de texto* que será provisto para su procesamiento.

(Aclaración del día 31 de mayo): el programa que tienen que realizar no debe tomar datos desde el teclado por ninguna razón, ni debe tener un menú de opciones (eso implicaría cargar por teclado la opción elegida). Tampoco debe haber en ninguna parte nada que solicite al usuario presionar una tecla para continuar. Todos los datos que el programa debe procesar (y solo esos datos) estarán en el archivo de texto cuyo formato y contenido se describe a continuación.

Es MUY IMPORTANTE que ese archivo se llame *peajes.txt* y que esté almacenado en la misma carpeta de proyecto que contiene al programa fuente. Si están haciendo pruebas con otros archivos (como *peaje25.txt* o *peaje75.txt*) no hay problemas, pero ASEGÚRENSE de que cuando entreguen el programa para su calificación, la función *open()* esté abriendo exclusivamente un archivo llamado *peajes.txt* Y NO OTRO.

El archivo de texto con los datos de entrada, consistirá de una primera línea (que aparecerá por única vez) en la que de alguna forma no estándar se indicará la fecha y la hora de creación de ese archivo¹. Y en alguna parte de esa línea, además, aparecerán dos caracteres de la forma “PT” (para indicar idioma portugués) o de la forma “ES” (para indicar idioma español). El idioma al que se hace referencia con esos dos caracteres le indica a los empleados de la empresa en qué idioma deben ser generados los reportes que se emitan, pero a los efectos de este trabajo, sólo impactará en un título. Algunos ejemplos (pero no los únicos posibles) de líneas de *timestamp* para este trabajo son los siguientes:

- ✓ 24 de mayo de 2023 – 13 hs 40 min – PT
- ✓ ES – 17:00 – 12/04/2022
- ✓ 09:00 – PT – 2021-10-05

En definitiva: una línea *timestamp* contendrá datos de fecha, hora e idioma, pero sin seguir un formato fijo ni estándar. Los dos grupos de caracteres “PT” y “ES” siempre aparecerán una y solo una vez, pero en cualquier lugar dentro de esa línea.

El resto de las líneas del archivo de entrada contendrá los datos de un vehículo, todos juntos conformando una cadena de 13 caracteres de largo (distribuidos como se explica más abajo), siempre con el mismo formato. Cada línea se separa de la que sigue con salto de línea.

Si se tienen registrados los datos de 20 vehículos que cruzaron cabinas Mercosur, entonces el archivo tendrá la línea de *timestamp* y luego otras 20 líneas, una por vehículo que haya cruzado. En cada línea aparecerán, estrictamente en este orden, los siguientes datos (que son los mismos que en la versión 1.0 se cargaban por teclado):

- **Caracteres del 0 al 6:** La patente. Si el vehículo es chileno el carácter en la posición 0(cero) será un blanco. De esa forma, en el archivo la patente siempre tendrá 7(siete) caracteres. Ahora, puede asumir que las patentes serán de 7 caracteres, y que no contendrán tampoco blancos (salvo el primer carácter de una patente chilena) ni letras minúsculas, ni caracteres diferentes de letras y números. Pero sigue siendo posible que incluso teniendo 6 o 7 caracteres podría no ser una placa válida para

¹ Una línea de este tipo en un documento, conteniendo fecha y hora de algún evento, se suele designar en el mundo informático como una línea de *marca temporal* (o *timestamp*).

ninguno de los seis países (y esto SÍ puede ocurrir en este trabajo). Y de todos modos, esa línea debe ser procesada.

- **Caracter 7:** un dígito entre 0 y 2 que indica el tipo de vehículo (0: motocicleta, 1: automóvil, 2: camión).
- **Caracter 8:** un dígito 1 o 2 que indica la forma de pago (1: manual, 2: telepeaje).
- **Caracter 9:** un dígito entre 0 y 4 que indica el país donde está la cabina que hizo el cobro (0: Argentina - 1: Bolivia - 2: Brasil - 3: Paraguay - 4: Uruguay). Note por favor, que el país donde está la cabina NO ES necesariamente el mismo país al que corresponde la patente. Y se asume también que todas las cabinas de peaje están en alguno de los cinco países mercosur (NO en Chile).
- **Caracteres 10 al 12:** un número entero de tres dígitos o menos, indicando la distancia en kilómetros que recorrió ese vehículo desde la última cabina de peaje que atravesó (asumimos que de alguna forma las cabinas se informan entre ellas esos datos). Aquí vendrá un cero para indicar que la cabina actual es la primera que ese vehículo atraviesa. Si la distancia es menor que 100, se indicarán ceros a la izquierda para completar las tres posiciones (obviamente, entonces, un cero se representará con tres ceros...)

Ejemplo: En el siguiente modelo mostramos qué forma podría tener un pequeño archivo de texto como el indicado:

```
2023/08/23 PT 04:30 PM
AX977WE123089
BCV8Q74210000
  CDFE24124100
    SD9999023050
      A8B9C9W111008
```

En el modelo la primera línea es la de *timestamp*, y los caracteres de idioma son “PT” (aproximadamente al centro de la línea).

Luego siguen 5 líneas (una por vehículo registrado).

- La primera es de una patente argentina, tipo de vehiculo 1, forma de pago 2, país de la cabina que hizo el cobro 3, y distancia 089 = 89.
- La segunda es de una patente brasileña, tipo de vehiculo 2, forma de pago 1, país de la cabina que hizo el cobro 0, y distancia 000 = 0.
- La tercera es de una patente chilena, tipo de vehiculo 1, forma de pago 2, país de la cabina que hizo el cobro 4, y distancia 100.
- La cuarta es de una patente *de otro país*, tipo de vehiculo 0, forma de pago 2, país de la cabina que hizo el cobro 3, y distancia 050 = 50.
- La quinta es de una patente *de otro país*, tipo de vehiculo 1, forma de pago 1, país de la cabina que hizo el cobro 1, y distancia 008 = 8.

Se pide desarrollar un programa que en base a todo lo anterior, procese los datos de un archivo con el formato indicado, usando un ciclo para levantar una por una las líneas del archivo, y muestre finalmente los siguientes resultados (y **estrictamente** estos resultados):

1. **Idioma** en el que se espera que los empleados de la empresa generen los reportes: Portugués o Español. Se aceptarán como válidas **solamente** las palabras “Español” (con ñ) o “Portugués” (con acento en la é), en cualquier combinación de minúsculas o mayúsculas (O sea, **debe** mostrar aquí **1(una)** cadena de caracteres con el idioma que corresponda).
2. **La cantidad de vehículos procesados de cada uno de los seis países**, y también **la cantidad de vehículos que NO eran de ninguno de esos seis** (O sea, **debe** mostrar aquí **7(siete)** contadores).

3. El **importe acumulado total** que ingresó a la empresa por todas las operaciones, considerando para este cálculo el importe final de cada ticket (O sea, debe mostrar aquí **1(un)** acumulador).
4. La **cantidad de veces** que apareció registrada en el archivo la primera patente leída, y cuál fue esa patente. (O sea, debe mostrar aquí **1(un)** contador y **1(una)** patente).
5. El **mayor importe final** que se haya pagado en un ticket, considerando todos los vehículos procesados, e indicando además la patente del vehículo al que se le hizo ese ticket (O sea, indicar aquí **2(dos)** resultados: **el mayor importe final**, y **la patente del vehículo** al que se le hizo ese ticket). Aclaración del día 29 de mayo: si el mayor importe aparece repetido varias veces dentro del archivo, se debe mostrar la patente del primer vehículo que tenga ese importe mayor.
6. El **porcentaje redondeado a no más de dos decimales** que representa la cantidad de patentes de otros países sobre el total de patentes procesadas (es **1(un)** resultado de tipo float: **el porcentaje redondeado a no más de dos decimales**).
7. La **distancia promedio redondeada a no más de dos decimales** recorrida por los vehículos con patente de Argentina que pasaron por cabinas brasileñas (es **1(un)** resultado de tipo float: **el promedio redondeado a no más dos decimales**). Aclaración del día 31 de mayo: si el promedio no puede calcularse, muestre como resultado un 0(cero) en este punto. No muestre un mensaje informando que el promedio no puede calcularse, ni permita que el programa se clave.

En la Ficha 11 y en las clases prácticas de materia, se irán desarrollando temas adicionales necesarios para el desarrollo completo de este trabajo (incluyendo la forma de procesar línea por línea un archivo de texto). Mientras tanto, los estudiantes pueden ir investigando por sus propios medios la forma de hacer lo que necesiten hacer. Y por supuesto, siempre pueden preguntar en clases o por medio del foro para consultas sobre el TP2.

Se proveerán a través de links del Aula Virtual cuatro archivos de texto (peaje25.txt, peaje50.txt, peaje75.txt, peaje100.txt) a modo de archivos de prueba para que los estudiantes puedan testear sus programas y validar lo que están haciendo. El primero contiene 25 líneas, el segundo contiene 50, el tercero 75 y el cuarto 100 (en todos los casos, además de esas líneas está la *timestamp* al principio)

Para el archivo *peaje25.txt*, los resultados que deberían obtener son los siguientes:

```
(r1) - Idioma a usar en los informes: Portugués

(r2) - Cantidad de patentes de Argentina: 3
(r2) - Cantidad de patentes de Bolivia: 3
(r2) - Cantidad de patentes de Brasil: 6
(r2) - Cantidad de patentes de Chile: 3
(r2) - Cantidad de patentes de Paraguay: 5
(r2) - Cantidad de patentes de Uruguay: 3
(r2) - Cantidad de patentes de otro país: 2

(r3) - Importe acumulado total de importes finales: 7289

(r4) - Primera patente del archivo: CA04615 - Frecuencia de aparición: 2

(r5) - Mayor importe final cobrado: 576 - Patente a la que se cobró ese importe: DH952XI

(r6) - Porcentaje de patentes de otros países: 8.0%

(r7) - Distancia promedio recorrida por vehículos argentinos pasando por cabinas brasileñas: 36.5km
```

Los resultados esperados para los otros archivos serán informados a los estudiantes con el correr de los días... Alentamos a todos a comenzar a trabajar y compartir con otros grupos mediante el foro o mediante consultas presenciales los resultados que están obteniendo, y discutir las posibles fallas para intentar descubrir errores y enmendarlos. Lo único que NO aconsejamos que hagan (obviamente) es lisa y directamente intercambiar programas fuente... Ya saben: eso ya sería hacer trampa descarada. Y hay gente allá afuera que vive de eso (como hemos alertado...)

Finalmente, aclaramos algo MUY importante: Este trabajo también será calificado en forma automática, pero ahora no solo será automática la asignación de entradas (vienen desde un archivo, así que ahí no hay nada que puedan tocar), sino que ahora también controlaremos las salidas en forma automática. Y para eso es absolutamente vital que respeten los nombres de las variables de resultados que más abajo les indicamos.

(Aclaración del 31 de mayo): ~~El orden de las salidas no es exigible, pero sería deseable que aparezcan en la forma indicada en la captura de pantalla donde hemos mostrado los resultados para el archivo *peaje25.txt*.~~ El orden de las salidas **es exigible**, tal cual como se muestran en la captura para el archivo *peaje25.txt* de la página anterior. Tampoco muestren nada más **en ninguna otra parte del programa**: Los únicos `print()` que el programa debe tener son los que están indicados en el enunciado para generar la secuencia de salidas pedidas. Y **nada más** que esos `print()` que se ven en la secuencia más abajo en esta misma página. No pongan títulos. No pongan mensajes de bienvenida. Ni saludos. Ni agradecimientos.

Por lo tanto, para facilitarles las cosas en cuanto a respetar los nombres de las variables y el orden **exigido sugerido** para las salidas, mostramos aquí la secuencia de instrucciones `print()` que sus programas deberían tener al final (sí, entendieron bien: copien y peguen al final de su código fuente toda la secuencia de salidas que sigue, y respeten a rajatabla **el orden de las salidas, el formato, los mensajes, la cantidad de líneas y** los nombres de las variables que nosotros hemos usado para almacenar los resultados y que figuran en color rojo en el modelo):

```
# Visualización de resultados...
print('(r1) - Idioma a usar en los informes:', idioma)

print()
print('(r2) - Cantidad de patentes de Argentina:', carg)
print('(r2) - Cantidad de patentes de Bolivia:', cbol)
print('(r2) - Cantidad de patentes de Brasil:', cbra)
print('(r2) - Cantidad de patentes de Chile:', cchi)
print('(r2) - Cantidad de patentes de Paraguay:', cpar)
print('(r2) - Cantidad de patentes de Uruguay:', curu)
print('(r2) - Cantidad de patentes de otro país:', cotr)

print()
print('(r3) - Importe acumulado total de importes finales:', imp_acu_total)

print()
print('(r4) - Primera patente del archivo:', primera, '- Frecuencia de
aparición:', cpp)

print()
print('(r5) - Mayor importe final cobrado:', mayimp, '- Patente a la que se cobró
ese importe:', maypat)

print()
print('(r6) - Porcentaje de patentes de otros países:', porc, '\b%')

print()
print('(r7) - Distancia promedio recorrida por vehículos argentinos pasando por
cabinas brasileñas:', prom, '\bkm')
```