# Algoritmos y Estructuras de Datos

# Trabajo Práctico 4 (2024)

## *Implementación de un Sistema Simple de Gestión de Envíos por Correo (y Códigos Postales) - Versión 4.0*

- **Introducción:**

El *Correo Central Argentino*, conociendo el éxito de las versiones anteriores, solicita ahora una versión 4.0 del programa que permita obtener estadísticas a partir de todas las transacciones del último mes.

En esta versión 4.0 volveremos a tener un archivo de texto con cierto volumen de datos ya grabados allí, pero ese archivo ahora tendrá formato *comma separated values* (*.csv*): cada línea del archivo estará compuesta por los distintos valores de un ticket separados por una coma, en lugar de venir todos agrupados en una gran cadena de formato fijo.

A diferencia de la versión anterior, en la versión 4.0 los datos deberán ser leídos desde ese archivo de texto, y luego deberán ser almacenados en un un archivo binario de registros, *sin generar un vector de registros/objetos en ese momento* (pero puede pedirse que generen un vector como parte de otros procesos). Toda la funcionalidad requerida debe ser programada directamente sobre el contenido de ese archivo binario (a menos que se especifique claramente otra cosa).

El programa debe ser gestionado a través de un menú de opciones. La corrección/calificación de este trabajo será realizada en forma manual por los profesores, así que no hay ahora restricciones en cuanto al formato y estilo de las salidas, ni hay tampoco exigencia en cuanto a cargar datos en cierto orden riguroso. Pero habrá un archivo de texto (con formato *.csv*) de entrada que deberán procesar.

Cada registro del archivo binario a crear debe mantener los datos de una transacción de envío postal. Por cada envío se asumen los mismos datos que para el TP3, pero quede claro que cada equipo de programadores puede decidir agregar nuevos atributos si los creen necesarios (lo que NO deben hacer, es **eliminar** algún atributo de los que son exigibles según la enumeración que sigue). Cada registro/objeto de tipo **Envio** deberá contener los datos siguientes:

- Código postal (CP) del envío (una cadena).
- Dirección física del destino (una cadena).
- Tipo de envío (un número entero entre 0 y 6).
- Forma de pago (un número entero (1: efectivo, 2: tarjeta de crédito)).

Al igual que en el TP3, los CP de Argentina y de los países vecinos de Argentina siguen un formato de acuerdo al mismo modelo indicado en las versiones de los TP anteriores. Y pueden venir CP de otros países (en cuyo caso deberán identificarse como “Otro” cuando sea requerido).

La forma de calcular los importes finales a cobrar por cada envío, están especificadas en los enunciados/requerimientos de las versiones de los TP anteriores (y por supuesto deben volver a aplicarse aquí en la versión 4.0).

La versión 4.0 del programa deberá basarse también en un menú de opciones. Pero a diferencia del TP3, la opción 1 de esta nueva versión debe tomar los datos de todos los tickets desde en un *archivo de texto envios-tp4.csv* que será provisto para su procesamiento con formato *.csv* como se explica a continuación.

El archivo de texto con los datos de entrada tendrá la primera línea de *timestamp* igual que en el TP2 y el TP3, y las líneas a partir de la segunda tendrán los datos de los tickets separados por una coma (terminando cada línea con un salto de línea). Estrictamente, la segunda línea tendrá los nombres o descriptores de los valores (y es una línea informativa que deberá ser ignorada), y a partir de la tercera vendrán los datos. El siguiente es un modelo de la forma que puede tener el archivo de entrada:

```
07:10 HC 2024-07-08
codigo_postal,direccion_fisica,tipo_envio,forma_pago
3740353,Revolucion 1357.,1,1
782639,Pacifico 987.,3,2
3283,Los Andes #999.,0,2
7X1274,Atlantico 987.,6,1
```

Note que este formato elimina el problema de tener que compensar con ceros a la izquierda los valores numéricos, o con blancos (a la derecha o a la izquierda) las cadenas. Además, con este formato el contenido del archivo está autodefinido (los nombres de los campos de la segunda línea expresan con cierta claridad qué es lo que contienen las líneas que siguen).

Se pide que esta versión del programa mantenga *la clase* **Envio**, con los atributos indicados aquí (más los que el equipo de trabajo determine que pueda necesitar), y a través de las opciones del menú **genere un archivo binario de registros**, de forma que cada registro de ese archivo contenga los datos de un envío. El programa debe incluir al menos los mismos dos módulos separados que en la versión anterior:

- En uno de esos módulos debe definirse la clase *Envio*, incluyendo los métodos o funciones separadas que el grupo determine que son aplicables.
- En el otro módulo, debe desarrollarse el programa principal, con un menú de opciones para realizar cada una de las tareas solicitadas más abajo.

Los procesos que deben estar disponibles en el menú principal son los siguientes. Note que en muchos casos se trata de adaptaciones de los que ya se pidieron para el TP3. **Y note también que en este TP4 no se requiere considerar el tipo de control de direcciones (HC o SC) en ningún momento y para ninguno de los requerimientos que siguen (en otras palabras, todos los registros deben ser procesados considerando que el tipo de control de SC y el formato de la dirección de envío no es relevante).**

1. Crear el **archivo binario** de registros de forma que contenga todos los datos de todos los envios guardados en el archivo de texto *envios-tp4.csv* que se provee junto con este enunciado. Cada vez que se elija esta opción, el archivo binario debe ser creado de nuevo desde cero, perdiendo todos los registros que ya hubiese contenido. Asegúrese de que antes de eliminar el viejo archivo, se muestre en pantalla un mensaje de advertencia al usuario de forma que tenga la opción de cancelar la operación. Repetimos: **NO DEBE CREAR UN ARREGLO DE REGISTROS/OBJETOS**, sino directamente pasar del archivo de texto al archivo binario. Y salvo en el punto 7 de este listado de procesos, EN NINGÚN OTRO PUNTO DEBE CREAR TAL ARREGLO, sino trabajar directamente con los datos contenidos en el archivo binario.
2. Cargar por teclado los datos de un envio, aplicando procesos de validación para cada campo, y agregar un registro con esos datos directamente al final del **archivo binario**. Cada vez que se elija esta opción, **el nuevo registro debe agregarse al final del archivo binario, sin perder ninguno de los registros que el archivo ya contenía**. Si el archivo no existiese, debe ser creado y luego agregar el registro cargado.
3. Mostrar **todos los datos de todos los registros del archivo binario**, tal como están grabados (sin ningún proceso de ordenamiento previo). Cada registro debe ocupar una sola línea en pantalla, **y debe mostrarse también el nombre del país al que corresponde el código postal**.
4. Mostrar **todos** los registros del **archivo binario** cuyo código postal sea igual a **cp**, siendo **cp** un valor que se carga por teclado. Al final del listado mostrar una línea adicional indicando cuántos registros se mostraron.

5. Buscar si existe en el **archivo binario** un registro cuya dirección postal sea igual a **d**, siendo **d** un valor que se carga por teclado. Si existe mostrar el registro completo. Si no existe indicar con un mensaje. La búsqueda **debe** detenerse al encontrar el primer registro que coincida con el criterio pedido.

6. Determinar y mostrar la cantidad de envíos de cada combinación posible entre tipo de envío y forma de pago en el **archivo binario**. Como son siete tipos de envíos posibles y son dos las formas de pago posibles, entonces se trata de 7 * 2 = 14 contadores, que obviamente deben ser gestionados en una matriz de conteo. Muestre solo los contadores cuyo valor final sea diferente de cero. Observación: ni siquiera se les ocurra plantear un esquema de 15 condiciones y 15 contadores separados… Esto se resuelve con una matriz de conteo o nada.

7. En base a la matriz que se pidió generar en el ítem anterior, muestre la cantidad total de envíios contados por cada tipo de envío posible, y la cantidad total de envíos contados por cada forma de pago posible. Es decir, se pide por un lado, totalizar las filas de esa matriz, y por otro, totalizar las columnas.

8. Recorrer el **archivo binario**, y calcular el importe promedio pagado entre todos los envíos que figuran en el archivo. Y ahora sí, generar en memoria un **arreglo de registros/objetos** con todos los envíos del archivo binario cuyo importe sea mayor al promedio que acaba de calcular. **Muestre el arreglo**, pero ordenado de menor a mayor de acuerdo al código postal. En este punto, los programadores deben considerar que la cantidad de datos en el vector podría ser realmente un número grande o muy grande, y por lo tanto, no deberían aplicar un método de ordenamiento simple. Tienen al menos el **Shellsort** explicado en clases. El archivo de entrada en formato .csv tiene 100000 (cien mil) líneas de datos, por lo que muy posiblemente el tamaño del "arreglito" que les tamos pidiendo ronde los 50000 (cincuenta mil) objetos... Deduzcan lo que deben hacer…