

Algoritmos y Estructuras de Datos
Trabajo Práctico 4 - 2025
Implementación de un Sistema Simple de Gestión de Órdenes de Pago (v4.0)

a.) Introducción

Luego de unas semanas de operación exitosa de la versión 3.0, la gerencia de la empresa de envíos decidió solicitar una *nueva* (¿última?) versión del software de gestión. Los algoritmos de cálculo se mantienen inalterables, las reglas expresadas anteriormente, también; para esta versión no hay cambios en el menú de opciones, ni en el *formato de salidas* solicitado, ni en la estructura del archivo de entrada (que será provisto nuevamente), ni en los atributos de un envío.

Será también provista a los alumnos la clase **Envío** ya desarrollada, con la función constructora y el método `__str__()` ya desarrollados (DEBEN importarla y usarla TAL CUAL se les provee, sin cambiar NADA de esa clase).

Pero hay **nuevos requerimientos** (o sea, nuevas salidas) que se expresan a continuación:

b.) Requerimientos

Opción 1: Cargar Envíos

Debe importar los envíos del archivo “envíos.csv” (que es exactamente el mismo que el usado en el TP3) y agregarlos a un vector, de tal manera que el vector se mantenga **siempre ordenado por identificación del destinatario**, de forma descendente.

Cuando haya terminado de cargar el arreglo, y antes de salir de la opción 1 del menú, debe solicitar al usuario que ingrese por teclado un **valor i** que hace referencia a un índice del arreglo que se ha cargado. Y con estos elementos, mostrar las siguientes salidas:

Resultados esperados (Opción 1):

- **r1.1:** El Identificador de pago de la orden que esté en el índice **i** del arreglo.
- **r1.2:** El identificador de pago de la orden que está en el índice **3i + 1** si **i** es impar o el índice **i // 2** si **i** es par. En caso de que este resultado estuviera fuera de rango, mostrar el último elemento del vector.

Opción 2: Generar Archivo

Crear un archivo binario que contenga todos aquellos envíos del vector cuya comisión supere al promedio de comisiones cobradas para su moneda de origen.

Resultados esperados (Opción 2):

- **r.2.1:** Listado (sin título) de los envíos almacenados en el archivo, a razón de un envío por línea de salida, y en el orden en el que están almacenados. Para mostrar cada registro, use exclusivamente el método `__str__()` tal como figura en la clase Envío que será provista.

Opción 3: Buscar Envío

Buscar si existe un envío para la identificación del destinatario **id**, siendo **id** un valor cargado por teclado. Si se encuentra un envío con este código de identificación, **modifique** el monto nominal de la orden, incrementando un 17% y redondeando el resultado a la centena más próxima. Si, por ejemplo, el monto nominal original fuese 1.000, el nuevo monto nominal sería 1.200. Si, en cambio, el monto nominal original fuese 950, el nuevo monto nominal sería 1.100.

Resultados esperados (Opción 3):

- **r.3.1:** El monto nominal (y solo el monto nominal) antes de la modificación (Si no se encontró el envío buscado, muestre 0).
- **r.3.2:** El monto nominal (y solo el monto nominal) luego de la modificación (Si no se encontró el envío buscado, muestre 0).

Opción 4: Mayores por combinación de monedas

Almacenar en una matriz el **Envío** con mayor monto final, para cada combinación posible de moneda origen / moneda de pago.

Resultados esperados (Opción 4):

- **r.4.1:** Mostrar para cada combinación posible el código (y solo el código) del envío almacenado en cada casillero.

c. Sobre las entradas y salidas

Se mantienen las consideraciones del enunciado anterior (TP3). En definitiva, el programa DEBE respetar lo siguiente:

1. Cuando se le solicite al usuario que elija una opción de menú, la misma solicitud debe indicarse con la leyenda: **"Ingrese opción: "**. No importa la forma en la que se presenten las opciones, pero el pedido de opción debe respetar ese mensaje en la consola. Por ejemplo:
 1. Cargar envíos
 2. Mostrar listado
 3. Buscar
 4. Mayores
 0. Salir
2. La **opción para salir** debe ser la opción **0**. Nuevamente, no importa el nombre con el que se muestre esta opción en el menú, cuando el usuario ingrese la opción 0, el programa debe terminar.

3. Los **resultados** deben **mostrarse en el orden solicitado**, y utilizando los dos puntos (":") para separar el nombre del resultado, del valor del resultado en sí. Si, por ejemplo, en la opción 1 se piden los resultados **r1.1** y **r1.2** y el programa calcula $r1.1 = 5475.34$ y $r1.2 = 3$, entonces la salida (Cuando se seleccione la opción 1) debe ser:

```
Ingrese opcion: 1
r1.1: 5475.34
r1.2: 3
```

4. Si se solicita mostrar un **arreglo** (unidimensional o bidimensional), entonces el programa **debe mostrar una línea por elemento del arreglo**, con la descripción del elemento antecedido por ":". Para el caso de arreglos bidimensionales, se debe respetar el recorrido por filas. Esto significa que, en un arreglo de 3 filas y 2 columnas, se mostraría primero fila 0, columna 0, luego fila 0, columna 1, luego fila 1, columna 0, y así sucesivamente. Evidentemente, si el enunciado pide que únicamente se muestren ciertos valores del arreglo, estos son los únicos que deben aparecer en la salida. Por ejemplo:

```
Origen ARS Destino ARS: 40
Origen ARS Destino USD: 11
Origen ARS Destino EUR: 33
Origen ARS Destino GBP: 29
Origen ARS Destino JPY: 9
Origen USD Destino ARS: 29
Origen USD Destino USD: 9
Origen USD Destino EUR: 14
Origen USD Destino GBP: 11
Origen USD Destino JPY: 29
Origen EUR Destino ARS: 38
Origen EUR Destino USD: 38
Origen EUR Destino EUR: 6
Origen EUR Destino GBP: 19
Origen EUR Destino JPY: 16
Origen GBP Destino ARS: 20
Origen GBP Destino USD: 30
Origen GBP Destino EUR: 7
Origen GBP Destino GBP: 28
Origen GBP Destino JPY: 29
Origen JPY Destino ARS: 6
Origen JPY Destino USD: 29
Origen JPY Destino EUR: 41
Origen JPY Destino GBP: 4
Origen JPY Destino JPY: 12
```