

Algoritmos y Estructuras de Datos
Trabajo Práctico 3 - 2025
Implementación de un Sistema Simple de Gestión de Órdenes de Pago (v3.0)

a.) Introducción

Debido a la gran demanda de transferencias que tiene la empresa los directivos decidieron ofrecer nuevas posibilidades en los envíos, permitiendo pagar el envío en una moneda y que el beneficiario lo reciba en otra moneda (si fuese requerido). La empresa sigue trabajando con las mismas 5 monedas que antes, pero ahora la moneda de origen y de destino no necesariamente serán las mismas, y existe una tasa de conversión entre una moneda y la otra. También se asignó un código numérico a cada moneda, quedando de la siguiente manera:

1: ARS, 2: USD, 3:EUR, 4:GBP, 5:JPY

Además, para facilitar el trabajo de quien opera el programa, se decidió que el mismo sea controlado por un menú, que cuenta con las siguientes opciones:

1. Cargar Envíos
2. Mostrar Resultados
0. Salir

Finalmente, en esta nueva versión del programa, se decidió cambiar el formato del archivo de entrada. En la versión previa, se utilizaba un formato de texto con campos de longitud fija, donde cada línea representaba una orden de pago. Ahora se optó por realizar el ingreso de datos desde un archivo con formato CSV, que contiene una línea por cada envío registrado, con los valores de los campos que se requieren para cada envío. Estos campos, que deberán ser modelados a través de una **clase Envío**, son:

1. **Código de la orden de pago:** Presenta un nuevo formato, ahora el código de la orden de pago tiene la forma:

<Código Moneda Origen>|<Código Moneda Pago>|<Identificador de pago>

Por ejemplo, si el código de orden de pago fuera **01|05|AB4F35**, la moneda de origen es 1 (ARS), la moneda de pago 5 (JPY) y un identificador generado para esa orden AB4F35.

2. **Identificación del destinatario:** una cadena que representa la identificación del destinatario.
3. **Nombre del destinatario:** una cadena que representa el nombre del destinatario.
4. **Tasa de conversión** (de moneda origen a moneda de pago): El factor por el que se debe multiplicar el monto en la moneda de origen para obtener el valor en la moneda de pago.
5. **Monto nominal a enviar.**
6. **Identificador del algoritmo de comisión.**
7. **Identificador del algoritmo impositivo.**

ACLARACIÓN IMPORTANTE: Cuando se haga referencia al monto final de una orden, se estará siempre esperando que este monto sea expresado en la moneda de pago. Esto significa: hacer los cálculos de comisiones e impuestos según aplique y luego convertir el monto obtenido a la moneda de pago utilizando el factor de conversión correspondiente. Por ejemplo, si la moneda de origen es ARS, y la de destino es JPY, se debe multiplicar el monto final en ARS por el factor de conversión a JPY.

Las tablas de los algoritmos son las mismas que en el TP1, y se detallan a continuación

| Tabla de Algoritmos para Cálculo de Comisiones (Cálculo del Monto Base) | |
|---|---|
| Número | Detalle |
| 1 | Moneda: ARS comision = 9% del monto_nominal. monto_base = monto_nominal - comision. |
| 2 | Moneda: USD Si el monto_nominal es menor a 50.000, entonces comision = 0. Si el monto_nominal es mayor o igual a 50.000, pero menor a 80.000, entonces comision = 5% del monto_nominal. Si el monto_nominal es mayor a 80.000, entonces comision = 7.8% del monto nominal. monto_base = monto_nominal - comision. |
| 3 | Moneda: EUR o GBP Se cobra siempre un monto fijo de 100 en esa moneda, por gastos (monto_fijo = 100). Si el monto_nominal es mayor a 25.000 (en esas monedas), entonces comision = 6% del monto_nominal. monto_base = monto_nominal - (monto_fijo + comision). |
| 4 | Moneda: JPY Si el monto nominal es menor o igual a 100.000, entonces comision = 500 en todos los casos. Si el monto_nominal es mayor a 100.000, entonces comision = 1.000 en todos los casos. monto_base = monto_nominal - comision. |
| 5 | Moneda: ARS Si el monto_nominal es menor a 500.000, entonces comision = 0. Si el monto_nominal es mayor o igual a 500.000, entonces comision = 7% del monto_nominal. Pero si la comisión fuese mayor a 50.000, entonces comision = 50.000. monto_base = monto_nominal - comision. |

| Tabla de Algoritmos para Cálculo Impositivo (Cálculo del Monto Final) | |
|---|---|
| Número | Detalle |
| 1 | Si el monto_base es menor o igual a 300.000, entonces impuesto = 0. Si el monto_base es mayor a 300.000, entonces: excedente = monto_base - 300.000 impuesto = 25% del excedente monto_final = monto_base - impuesto. |
| 2 | Si el monto_base es menor a 50.000, entonces impuesto = 50 en todos los casos. Si el monto_base es mayor o igual a 50.000, entonces impuesto = 100 en todos los casos. monto_final = monto_base - impuesto. |
| 3 | impuesto = 3% del monto_base para todos los casos. monto_final = monto_base - impuesto. |

b.) Requerimientos

Opción 1: Cargar Envíos

En la primera opción del menú, el programa debe procesar el archivo csv provisto con este enunciado, línea por línea, y cargar cada uno de los envíos en un vector de envíos. Al finalizar este procesamiento, siempre como parte de la opción 1, el programa debe mostrar los siguientes resultados:

- **r1.1:** Cantidad de envíos leídos y cargados en el vector.
- **r1.2:** Cantidad de envíos leídos donde la moneda de origen es diferente a la moneda de pago.

Nota: NO SE PIDE mostrar cada uno de los envíos. Simplemente se deben mostrar estos dos resultados de control.

Opción 2: Mostrar Resultados

En esta opción se debe utilizar el vector cargado en la opción anterior para entregar los siguientes resultados:

- **r2.1:** Porcentaje promedio que se cobró en concepto de comisiones (porcentaje del monto de las comisiones calculadas, respecto al monto nominal). Por ejemplo, si la comisión es \$100 sobre una orden con monto nominal de \$1.000, la comisión para esa orden fue del 10%.
- **r2.2:** El identificador de pago de la orden que tuvo mayor porcentaje de descuento (considerando comisiones e impuestos).
- **r2.3:** Monto final correspondiente a la orden del resultado r2.2.

b.) Sobre las entradas, salidas y otros requerimientos

El TP3 será evaluado usando un programa de control tipo runner, en forma similar a lo hecho para los TPs anteriores. Por lo tanto, es **MUY IMPORTANTE** que el programa que presenten respetar las exigencias siguientes:

1. El archivo csv que se provee como entrada, se llama “envios.csv”. Cuando usen la función `open()` para abrir ese archivo, debe asegurarse que **ese** (y no otro) sea el nombre del archivo enviado a la función. Un nombre diferente, provocará un error en la ejecución del runner, y penalización de puntos en la calificación.
2. Mínimamente, el programa debe contar con dos módulos. Uno para la definición de la clase y sus métodos asociados, y otro para el programa principal (conteniendo al menú). Que se entienda: el archivo de código fuente que contiene al programa principal, **DEBE** llamarse ***principal.py***: otro nombre, podría provocar un error en el runner.
3. Todo módulo que entreguen (que contenga un script ejecutable, incluido el módulo principal) **DEBE** tener control de ejecución mediante la variable `__name__`.
4. 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:

```
=== MENÚ ===  
1) Cargar Envíos  
2) Mostrar Resultados  
0) Salir  
  
Ingrese opción:
```

5. 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.
6. Los **resultados** deben **mostrarse en el orden solicitado**, y utilizando los dos puntos (“:”) para separar el texto aclaratorio 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 opción:1  
r1.1: 5475.34  
r1.2: 3
```

7. Si se solicita mostrar un **arreglo** (unidimensional o bidimensional), entonces el programa **debe mostrar una línea por cada elemento del arreglo**, con la descripción del elemento terminando en “:”. Para el caso de arreglos bidimensionales, se debe respetar el recorrido por fila creciente. 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
```

Requerimiento Extra: Idea general para todos los cursos

Se agrega un resultado extra en la opción 2 (mostrar resultados), figurando como **r.2.4**. El enunciado de este extra se les hará visible en el día que corresponda a la clase práctica del curso en el que están inscriptos.

Cada curso deberá resolver el extra que se le indique en ese momento, **y solo ese resultado extra** (si llegasen a tener conocimiento por cualquier razón de los extras de otros cursos, no deben resolverlos ni entregarlos). Por supuesto, deben respetar el formato de salidas descrito en la consigna general de este trabajo práctico.