

Algoritmos y Programación I (75.40)

Trabajo práctico n.º 2

Primer cuatrimestre 2016

Consigna

Implementar un sistema de análisis de precios de productos en supermercados. El sistema debe tomar como entrada los datos a partir de un conjunto de archivos CSV¹, y brindar una interfaz interactiva para obtener diferentes tipos de reporte.

Requerimientos

El programa debe cargar en memoria los datos proporcionados en los archivos CSV. Si alguno de los archivos no existe o no puede cargarse por alguna razón (por ejemplo, si alguna de las líneas del archivo no respeta el formato), se espera que el programa muestre un mensaje de error apropiado y, si es fatal², finalice la ejecución.

Luego debe presentar un menú de opciones que permita:

1. Calcular la inflación por supermercado
2. Calcular la inflación en un producto determinado, discriminado por supermercado

¹ <http://es.wikipedia.org/wiki/CSV>

² Un error fatal hace imposible continuar la ejecución. En cambio, por ejemplo, si uno de los productos no tiene descripción, es un problema, pero no evita que el programa se siga ejecutando.

3. Calcular la inflación general promedio
4. Obtener el supermercado con mejor precio para un producto determinado

Archivo CSV de supermercados

Uno de los archivos que el programa debe cargar al inicio es `supermercados.csv`. La primera línea del archivo contiene la cabecera:

```
id,nombre
```

El resto del archivo contiene un supermercado por línea, siguiendo el formato descrito en la cabecera. Por ejemplo:

```
id,nombre
1,Coto
2,Jumbo
3,Carrefour
```

Archivo CSV de productos

El archivo `productos.csv` es similar al de supermercados, describiendo un producto por línea. Por ejemplo:

```
id,nombre
1,Galleta arroz integral c/sal dos hermanos bol 100 grm
2,Galletitas express paq 108 grm
3,Galletitas dulces tita 36 uni est 648 grm
4,Galletitas dulce okebon con lunchera 705 grm
5,Galletitas rellenas pepitos vainilla paq 384 grm
6,Galletas marinera la bizantina con gluten bsa 180 grm
7,Galletitas de avena quaker c/ chispas de chocolate paq 264 grm
```

Archivo CSV de precios

El archivo `precios.csv` describe el precio de cada producto por cada supermercado y por cada mes. Por ejemplo:

```

id_supermercado,id_producto,periodo,precio
1,1,201601,18.5
1,1,201602,19.31
1,1,201603,19.67
1,1,201604,20.3
1,1,201605,20.79
1,1,201606,21.75
1,1,201607,21.96
1,1,201608,22.73
1,1,201609,23.55
1,1,201610,23.81
1,1,201611,24.73
1,1,201612,24.95

```

En este ejemplo, la primera línea (1,1,201601,18.5) indica que el supermercado con id 1 (Coto) ofrecía el producto con id 1 (Galleta arroz integral...) en enero de 2016 a \$18,5.

Menú de opciones

Una vez finalizada la carga de datos en memoria, el programa debe mostrar un menú de opciones para permitir al usuario obtener los distintos tipos de reporte. Por ejemplo:

```

Menu principal
-----
1. Inflación por supermercado
2. Inflación por producto
3. Inflación general promedio
4. Mejor precio para un producto
5. Salir
Opción:
-----

```

A continuación describe cada una de las opciones del menú. En todos los casos se espera que el programa valide los datos ingresados por el usuario, mostrando un mensaje de error apropiado en caso de ser necesario.

Inflación por supermercado

Cuando el usuario seleccione esta opción, el sistema deberá preguntarle para qué período de tiempo desea calcular la inflación. El usuario deberá ingresar dos pares de valores para el año y mes.

El programa deberá calcular y mostrar la inflación de precios de cada supermercado en el período indicado, como el promedio de la inflación de cada uno de los productos producto, y para calcular la inflación de un producto utilizará la fórmula:

$$inflación = 100 \frac{precio_{final} - precio_{inicial}}{precio_{inicial}}$$

Al finalizar el cálculo se debe imprimir la inflación de cada supermercado. Por ejemplo:

Coto: 13.11%
Jumbo: 20.15%
Carrefour: 16.85%

Inflación por producto

El programa preguntará al usuario el período de tiempo (al igual que en el caso anterior) y también el nombre de un producto.

No será necesario que el usuario ingrese el nombre exacto del producto; se espera que el programa sea suficientemente inteligente y permita al usuario elegir entre productos cuyo nombre sea similar al texto ingresado. El diseño del flujo de selección de producto es parte de la evaluación. El programa debe preguntar si ese es el producto deseado antes de continuar. Además debe manejar correctamente el caso de que no haya ningún producto con nombre similar. Por ejemplo:

Ingrese el nombre de un producto: koebon

No se encontró ningún producto con nombre "koebon"

Ingrese el nombre de un producto: okebon

[Galletitas dulce okebon con lunchera 705 grm]

¿Este es el producto deseado? [s/n]:

Una vez seleccionados el período y el producto, el programa calculará la inflación de cada supermercado, de forma similar al caso anterior, solo que tendrá en cuenta únicamente el producto seleccionado.

Inflación general promedio

El programa debe preguntar al usuario el período de tiempo y luego mostrar la inflación general, calculada como el promedio de la inflación de todos los productos.

Mejor precio para un producto

El programa debe pedir al usuario un par de valores año y mes, y el nombre de un producto. Luego debe mostrar cuál es el supermercado que ofrece el mejor precio para ese producto, y el precio.

Criterios de aprobación

A continuación se describen los criterios y lineamientos que deben respetarse en el desarrollo del trabajo.

Informe

El informe debe consistir en una descripción del **diseño** del programa.

Debe recordarse que la etapa de diseño es *anterior a la implementación*, por lo tanto debe describirse, utilizando texto y/o diagramas, cómo se va a estructurar el código para cumplir con las especificaciones de la consigna.

Algunas preguntas que deberían responderse:

- A grandes rasgos, ¿cómo será el flujo del programa? ¿Cuáles archivos se leen o escriben, y en qué momentos?
- ¿Qué estructura(s) de datos se usan para representar los datos en memoria?
- ¿Cómo se calculan los reportes?

Código

Además de satisfacer las especificaciones de la consigna, el código entregado debe cumplir los siguientes requerimientos:

- El código debe ser claro y legible.
- El código debe estructurarse en funciones y, cuando corresponda, módulos. Las funciones deben definirse de la manera más genérica posible.
- Todas las funciones deben estar adecuadamente documentadas, y donde sea necesario el código debe estar acompañado de comentarios.

Entrega

La entrega del trabajo consiste en:

- El informe y código fuente impresos. Para el código fuente utilizar una tipografía `monoespacio`.
- El informe en formato *PDF*.
- Una versión digital de todos archivos `.py` de código, separados del informe. En el caso de ser más de un archivo, comprimidos en un `.zip`.

El informe impreso debe entregarse en clase. Los dos últimos (PDF y código fuente) deben enviarse a la dirección electrónica `tps.7540rw@gmail.com` con el asunto “*TP2 - PADRÓN1 - PADRÓN2*”.

Este trabajo práctico se desarrolla en forma **grupal**. El plazo de entrega vence el **lunes 9 de mayo de 2016**.