T.P. 2

Autores:

* Santos, Ramiro
* Wu, Nicolás

Cátedra: Wachenchauzer – Bárbara

Fecha de entrega: 9 de Mayo de 2016

Docente a cargo de la corrección: Diego Essaya

1. Estructuración del programa

Antes de comenzar decidimos dividir el trabajo en tres partes fundamentales:

-Guardar la información de los archivos en la memoria para poder trabajar con ella (manipulación de información de gran magnitud).

-Las 4 funciones que calculan la inflación por producto, por supermercado, la general y el mejor precio para un producto determinado.

-Un buscador que busque, valga la redundancia, el producto ingresado por el usuario.

2. Resolución

Primero lo que pensamos hacer es recorrer cada archivo y guardar cada línea en una lista, obteniéndose tres listas (una para cada archivo) y pudiendo así trabajar con cada archivo por separado. Luego, para poder manejar todos estos datos en conjunto, pensamos en ir guardando tanto el id de los supermercados, como el de los productos, junto con su precio en una determinada fecha y un determinado supermercado en una gran lista de listas, pero nos dimos cuenta que para obtener un dato de esta lista deberíamos recorrerla con un for, haciendo una búsqueda lineal, lo cual tardaría demasiado (teniendo en cuenta que el archivo de productos tiene como más de setenta mil líneas). Por lo tanto decidimos que es mejor guardar toda la información necesaria en un diccionario de diccionarios, de modo que quede: {Supermercado1: {Producto1:{Fecha1:{precio}………..  
Fechafinal:{precio}}…….Productofinal:{….}}…..Supermercadofinal:{……}}. Esto facilita el trabajo ya que, a diferencia de la búsqueda en listas, en un diccionario uno puede ‘meter la mano y agarrar lo que busca’, no hace falta recorrerlo todo, y así el programa no se lentifica.

Una vez hecho este diccionario, la realización de las funciones que calculan lo pedido por el usuario será más fácil, salvo las que le piden al usuario que ingrese un producto, el cual se utilizará para calcular lo pedido. Para estos casos deberemos hacer un buscador de productos.

Como bien dice la consigna, al ingresar el usuario un producto, por más que no escriba su nombre completo, el programa debe mostrarle todas las opciones que contienen la cadena que ingresó y preguntarle cuál es la que busca. Para ello lo que podemos hacer es recorrer la lista que contendrá todas las líneas del archivo productos (lista mencionada anteriormente) y verificar si lo ingresado por el usuario se encuentra en cada línea. Cada coincidencia se mostrará por pantalla asignada con un número, pudiendo luego el usuario indicar el número al que corresponde el producto que busca. Hemos pensado para la realización de este buscador que no se debe tener en cuenta si el usuario no ingresa nada o simplemente un espacio (puesto que no tiene sentido que busque eso y se le muestren todos los productos, que se debería a que la cadena espacio está en todos los nombres de los productos), y que sí se debe tener en cuenta que, a la hora de ingresar un nombre, este puede empezar tanto con minúscula como con mayúscula. Si esto no se tuviese en cuenta, al poner el usuario ‘galletitas’ por ejemplo solo se mostrarían los productos que tienen el nombre ‘galletitas’, mientras que unos cuántos empiezan con el nombre ‘Galletitas’; y es más que probable que el usuario al buscar ‘galletitas’, quiera encontrarse con todas las opciones y no sólo con las que no tienen mayúscula al inicio.

El resto del programa será: el menú principal, donde se ofrecen las opciones de cálculos que puede obtener el programa, validaciones, como por ejemplo si una fecha está bien escrita o un número está dentro del rango pedido, verificaciones, como la de ver si una fecha ingresada se encuentra entre los datos que el archivo productos ofrece, y por último y no por eso menos importante, las excepciones, que de hecho tienen una gran importancia. Estas se ubicarán en la función principal y explicarán al usuario el error (si es que hubo alguno) dependiendo de lo que haya sucedido (como por ejemplo si se abre un archivo que no existe o se modifican las fechas (que están en números) por letras).

3. Flujo del programa principal

El diagrama que se puede observar en la página siguiente representa el flujo del programa principal. Las excepciones no se diagramaron para una mejor visualización del esquema, pero hay que tener en cuenta que están y que si llega a saltar un error estas lo atraparán, interrumpiendo el programa y enviando un mensaje aclarativo al usuario.



***Diagrama de flujo***