

## Trabajo Práctico Nº 3

### Enanito's verdulería



Fecha presentación	16/11/2023
Fecha entrega	28/11/2023

## 1. Introducción

Blancanieves y los enanitos tuvieron tanto éxito vendiendo sus cosechas que necesitan un nuevo sistema para llevar cuenta de los pedidos que tienen. Nos contrataron para que les demos una mano haciendo un sistema de archivos para controlar las ventas.

## 2. Objetivo

El presente trabajo práctico tiene como objetivo que el alumno:

- Manejo correcto de archivos.
- Se familiarice con y utilice correctamente el lenguaje de programación Python.
- Desarrolle una comunicación amigable y entendible con el usuario.

Por supuesto, se requiere que el trabajo cumpla con las buenas prácticas de programación profesadas por la cátedra.

Se considerarán críticos la modularización, reutilización y claridad del código.

## 3. Enunciado

El sistema se basará en una serie de comandos que el usuario puede utilizar para guardar, modificar, eliminar y mostrar los pedidos de Blancanieves, los cuales estarán guardados en un archivo de tipo csv, y por otro lado se tendrán los clientes de cada pedido en otro archivo csv.

Un pedido está compuesto por:

- El id para identificar el pedido (se genera cuando se crea el pedido, en modo ascendente)
- El tipo de verdura que se quiere pedir
- La cantidad de verduras

En el archivo los pedidos se guardan con el formato

```
1 id_pedido;verdura;cantidad
2
3 Por ejemplo
4 2;T;6
```

Cabe aclarar que solamente se deben guardar en el archivo datos correctos: la cantidad debe ser positiva y las verduras sólo pueden ser tomate (T), brócoli (B), zanahoria (Z) o lechuga (L). Por otro lado, puede haber un pedido con más de un tipo de verdura, para lo cual se los identificará con el mismo número de id del pedido. Por ejemplo si se hace un pedido de 4 tomates y 5 zanahorias habrá dos entradas en el archivo de la siguiente forma:

pedidos.csv

```
1 7;T;4
2 7;Z;5
```

Además, se guardarán en otro archivo csv los clientes a los cuales hay que entregarles el pedido con el siguiente formato:

clientes.csv

```
1 id_pedido;nombre cliente
2
3 Por ejemplo:
4 7;Tomi
```

Entonces sabemos que el pedido de los 4 tomates y 5 zanahorias lo hizo Tomi porque las entradas del archivo tienen el mismo id de pedido.

### 3.1. Agregar

Para usarlo, el usuario deberá ingresar la palabra **agregar** seguida de los datos del pedido separados por un espacio. Es decir, si se quiere agregar un pedido en nombre de Agustina de 3 brócolis:

```
1 agregar 3 B Agustina
```

Si alguno de estos campos no se ponen en el comando, el pedido no debe crearse y debe haber un mensaje para avisarle al usuario de su error.

### 3.2. Modificar

Para usarlo, el usuario deberá ingresar la palabra **modificar** seguida del id del pedido (si el pedido con ese id no existe, se le debe indicar con un mensaje), y luego además poner la cantidad de verdura a agregar al pedido seguido del tipo de la misma.

Para modificar el primer pedido, agregándole 5 tomates:

```
1 modificar 1 5 T
```

Si se puede modificar correctamente, se debe avisar por pantalla.

Cabe aclarar que si ya hay una entrada en el archivo de la misma verdura para ese pedido, esta entrada debe ser modificada (reemplazando el número de la cantidad por el recientemente ingresado). Por otro lado, si no existe una entrada de esa verdura para ese pedido se tiene que agregar.

### 3.3. Eliminar

Para usarlo, el usuario deberá ingresar la palabra **eliminar** seguida del id del pedido a cambiar. En caso de que no exista, se le debe indicar que no existe.

```
1 eliminar 1
```

Para eliminar correctamente se debe eliminar TODAS las líneas con el id del pedido de ambos archivos, clientes.csv y pedidos.csv

### 3.4. Listar

Este comando muestra todos los pedidos por pantalla con un formato amigable para el usuario.

Para usarlo, se deberá ingresar la palabra **listar** seguida del id del pedido que se quiere listar, o si no ingresa ningún parámetro, se listan todos los pedidos.

Si se ingresa un id incorrecto (negativo o que no existe), se le debe mostrar un mensaje de error al usuario.

```
1 # muestra todos los pedidos
2 listar
```

```
1 # muestra el pedido 3
2 listar 3
```

## 4. Resultado esperado

Se espera que se creen las funciones y procedimientos para que el sistema se desarrolle con fluidez. Se debe tener una buena comunicación con el usuario, avisándole de sus errores e indicando qué cambios se van haciendo en el archivo y sus errores. Así mismo, se espera que se respeten las buenas prácticas de programación.

**ACLARACIÓN** Los comandos tienen que realizarse con argumentos de la línea de comando. No puede haber ningún input dentro del programa a entregar.

**IMPORTANTE!** Se deberá validar que los comandos reciban los parámetros necesarios, y en caso de no recibirlos, se deberá avisar al usuario y no realizar ninguna acción.

## 5. Entrega

El trabajo práctico debe ser realizado en un archivo llamado verduleria.py.

Por último debe ser entregado en la plataforma de corrección de trabajos prácticos **AlgoTrón** (patente pendiente) y será probado de forma manual por los correctores.

Para la entrega en **AlgoTrón** (patente pendiente), recuerde que deberá subir un archivo **zip** conteniendo únicamente los archivos antes mencionados, sin carpetas internas ni otros archivos. De lo contrario, la entrega no será validada por la plataforma.