AD

Álvaro Manuel Navarro Cruz

2º DAM

04/11/2024

Práctica 1

PERSISTENCIA EN FICHEROS

Índice

[Introducción (↑) 3](#_Toc181401037)

[Enlace Repositorio GitHub 3](#_Toc181401038)

[1.- Clase ArticuloAComprar (↑) 4](#_Toc181401039)

[2.- Clase ListaDeLaCompra (↑) 5](#_Toc181401040)

[3.- Clase Principal (↑) 7](#_Toc181401041)

[4.- Pruebas de ejecución (↑) 8](#_Toc181401042)

[5.- Valoración Personal 12](#_Toc181401043)

[6.- Bibliografía 13](#_Toc181401044)

Introducción [(↑)](#indice)

Para esta práctica se pide diseñar y codificar un modelo que permita gestionar listas de la compra adaptadas a los productos disponibles en una tienda.

El modelo deberá constar de 2 clases, a saber, ArtículoAComprar y ListaDeLaCompra, además de una clase Principal que haga uso de los métodos necesarios para leer un archivo de texto, escribir en él, modificarlo eliminando o agregando artículos y finalmente guardar los cambios en el mismo archivo de texto.

Para esta práctica se utilizará un archivo de texto “.txt”.

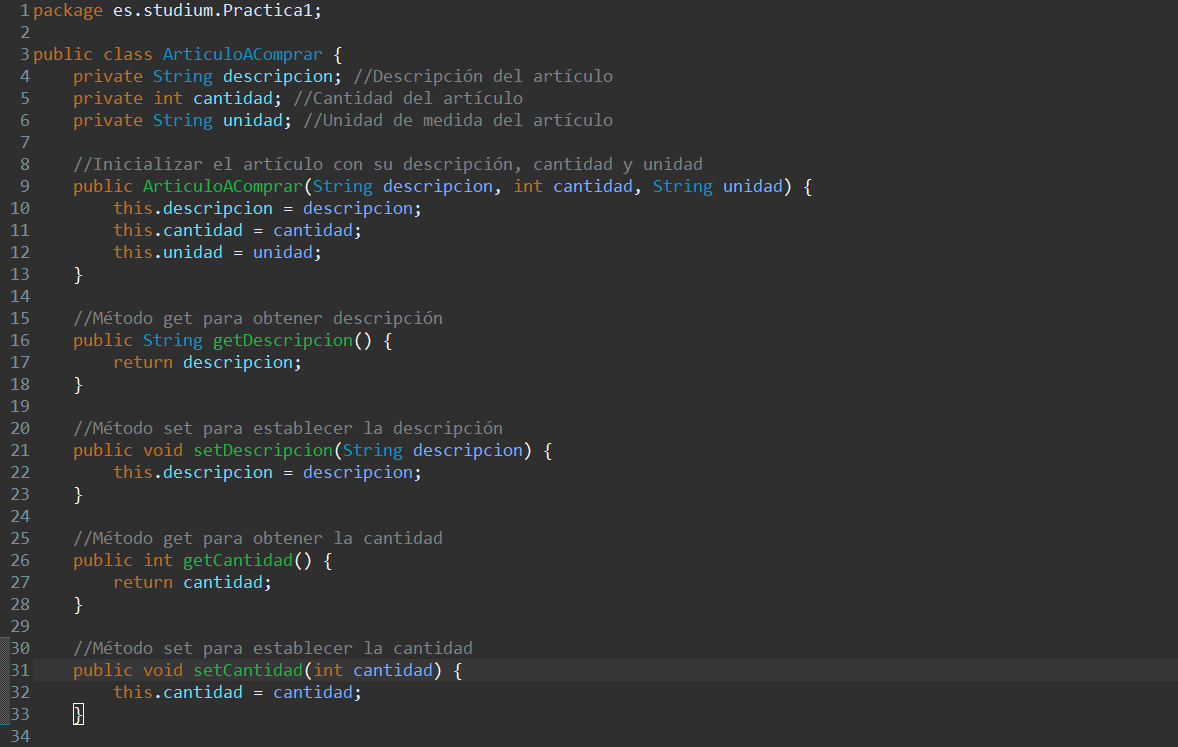
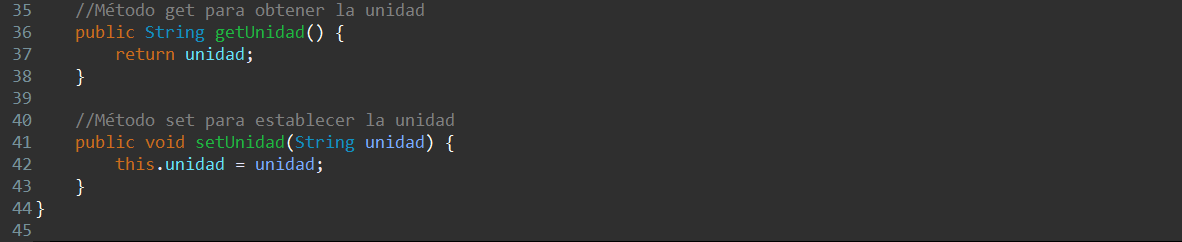
**Se explicará el funcionamiento del código a modo de comentarios en el mismo, desarrollando en cada función una breve explicación de lo que se realiza.**

Se **agregarán métodos adicionales a los proporcionados** en la clase ListaDeLaCompra con **objetivo** de hacer **más legible la clase Principal**, agregando métodos para cargar la lista, escribir sobre ella, guardarla y mostrarla.

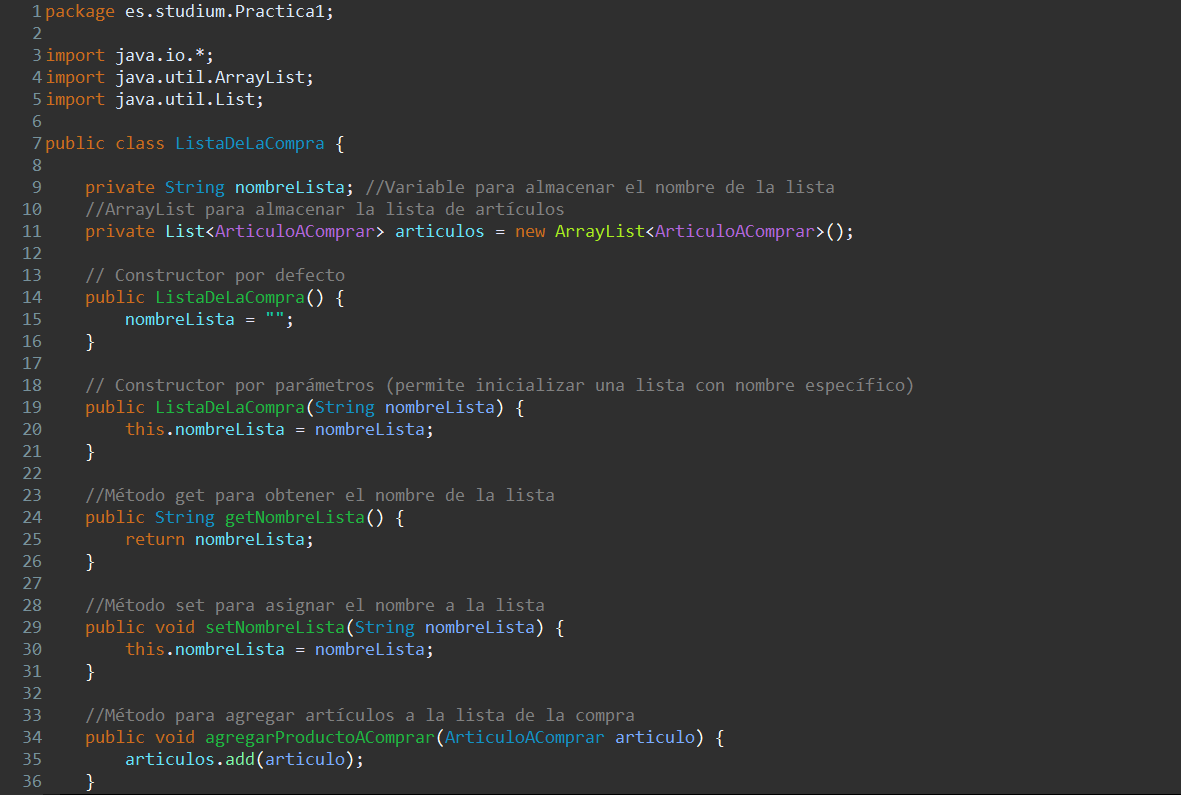
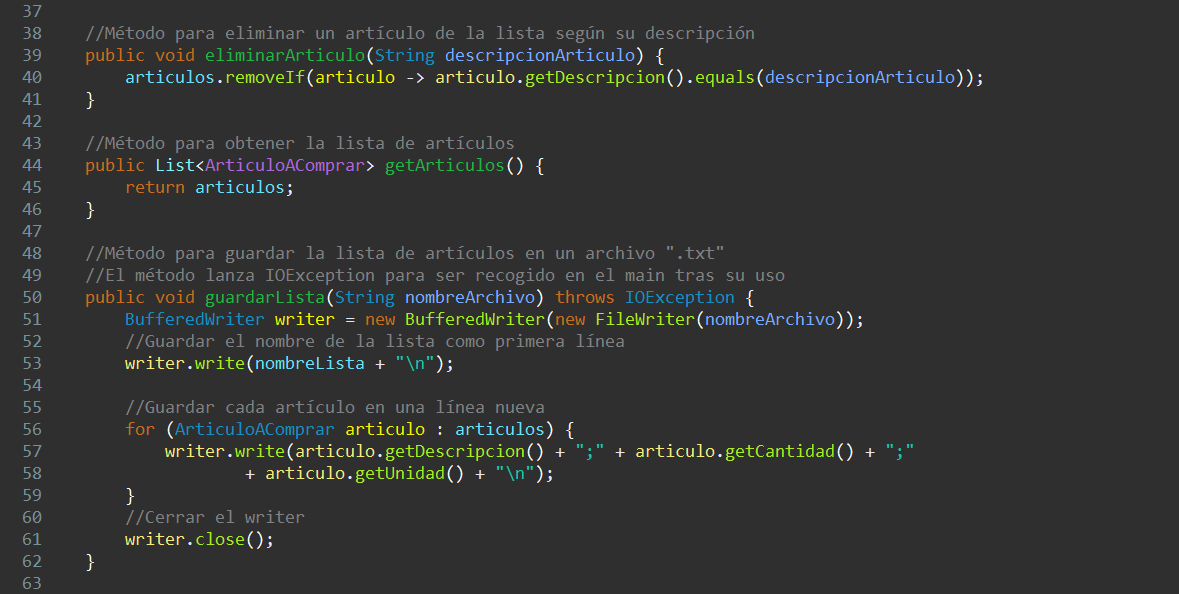
## Enlace Repositorio GitHub

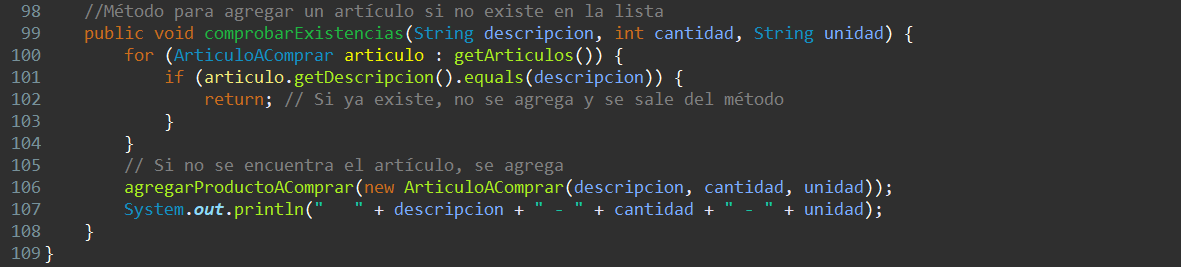
<https://github.com/AlvaroMfco/AD_Practica1.git>

1.- Clase ArticuloAComprar [(↑)](#indice)

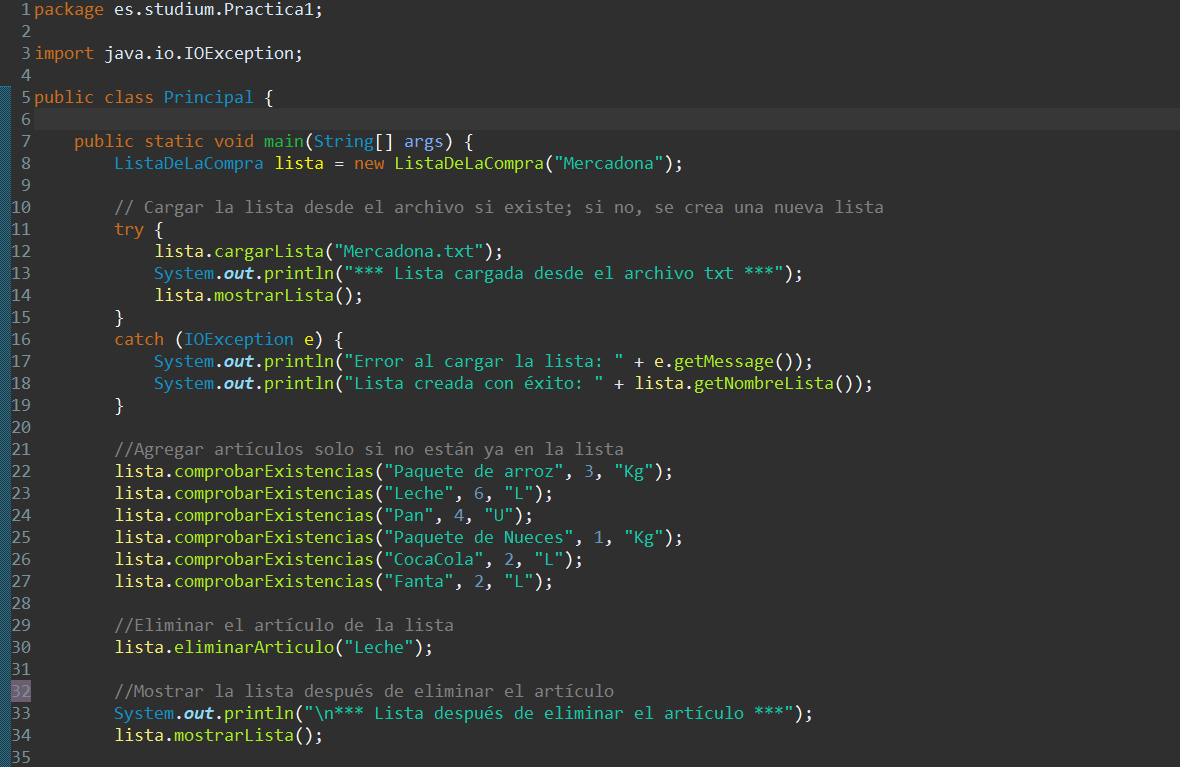
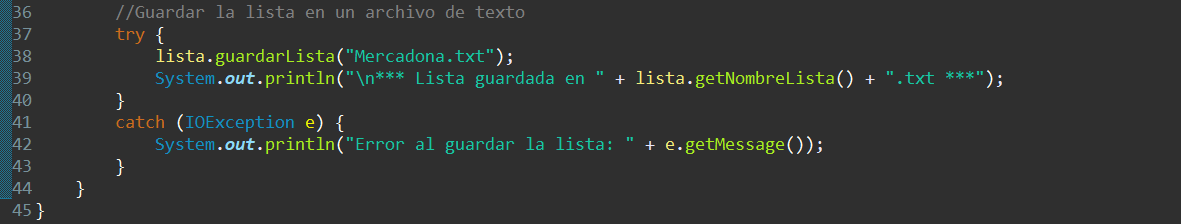


2.- Clase ListaDeLaCompra [(↑)](#indice)





3.- Clase Principal [(↑)](#indice)



La clase principal está exenta de métodos innecesarios, habiendo sido creados en la clase ListaDeLaCompra, de manera que queda mucho más legible.

En esta clase se crea en primer lugar una instancia de ListaDeLaCompra, en segundo lugar se procede a cargar la lista desde un archivo de texto mediante lista.cargarLista(“Mercadona.txt”). Si el archivo existe y se carga correctamente, se muestra por consola un mensaje de feedback y se muestra la lista con lista.mostrarLista();

Si el archivo no se puede cargar (porque no exista), se lanza una excepción con el informe y se nos indica que se ha creado una nueva lista.

Finalmente agregamos productos a la lista si no están ya añadidos, realizamos una eliminación de artículo mediante lista.eliminarArticulo() y se vuelve a mostrar la lista actualizada.

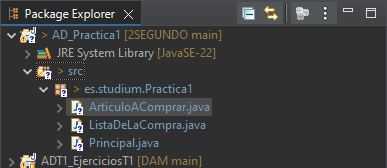
Tras estos pasos, se guardarán los cambios en “Mercadona.txt”.

4.- Pruebas de ejecución [(↑)](#indice)

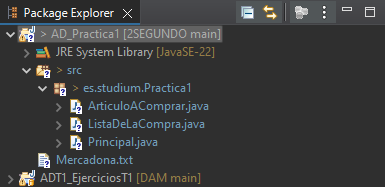
Para realizar las pruebas, primero vamos a comentar la línea para eliminar productos y vamos a comprobar nuestro txt generado:



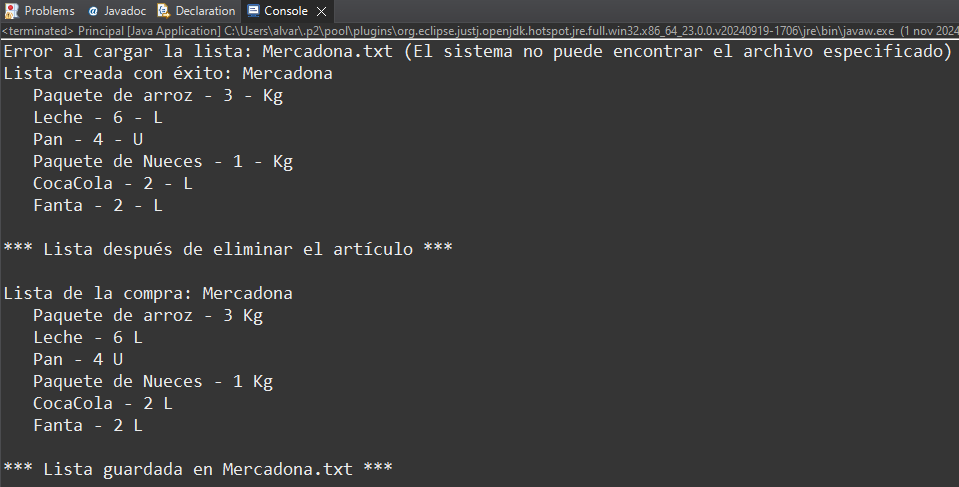
Nuestro explorador de paquetes se verá de esta manera antes de ejecutar el programa:



Una vez ejecutado, refrescamos el proyecto y aparecerá el txt creado:

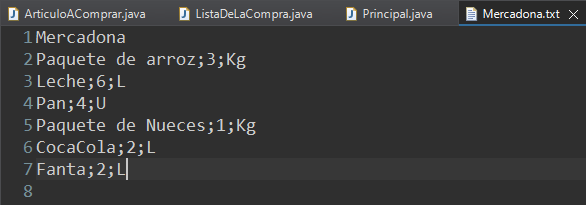


Por consola, obtendremos el siguiente resultado al ejecutarlo por primera vez:



El primer mensaje indicará que no se ha podido encontrar el archivo txt, puesto que aún no está creado.

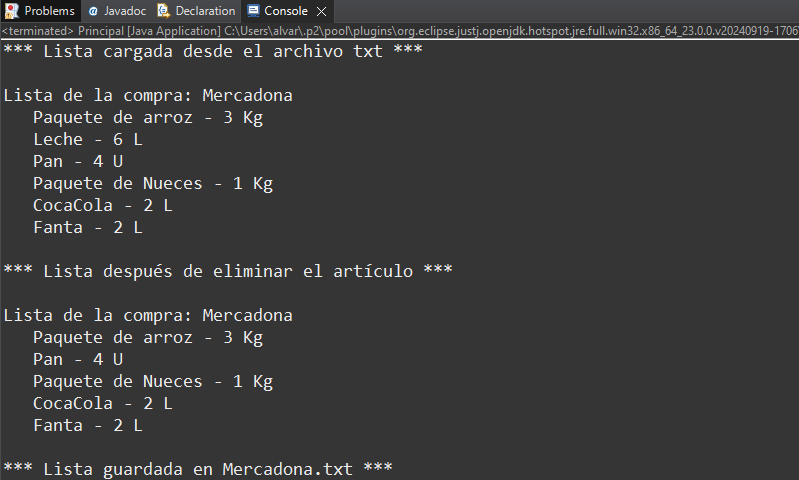
Si accedemos al archivo txt que se ha generado, veremos los productos añadidos:



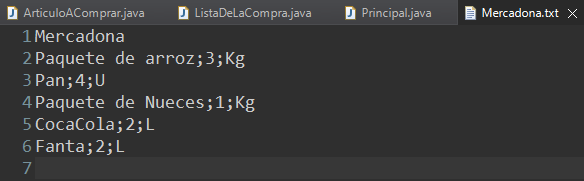
Ahora vamos a des comentar la línea para eliminar artículos y vamos a borrar el artículo “Leche”:



Ejecutamos y saldrá lo siguiente por consola:



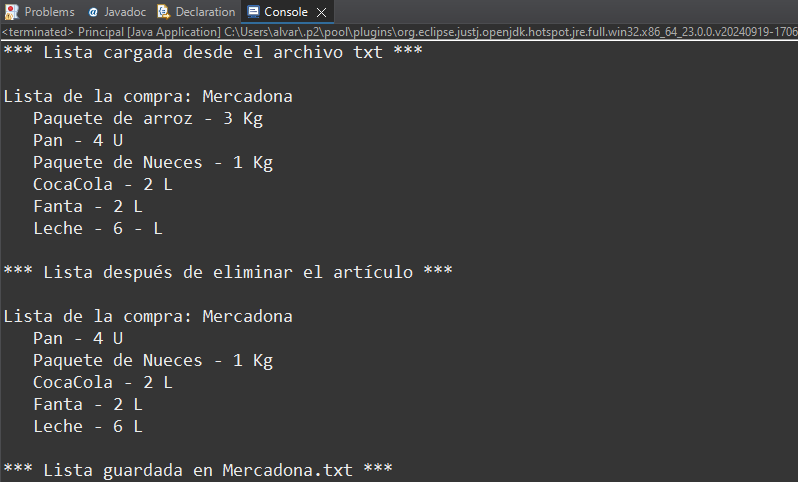
Si comprobamos el archivo Mercadona.txt, debería haber desaparecido el producto “Leche”:



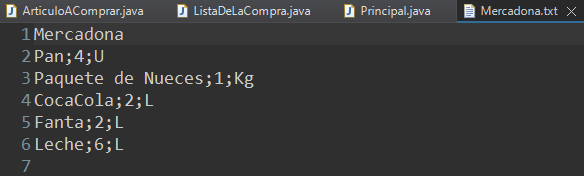
Vamos a seguir eliminando productos, por ejemplo, “Paquete de arroz”:



Por consola, aparecerá lo siguiente:



Si comprobamos el archivo Mercadona.txt, habrá desaparecido el artículo “Paquete de arroz”:



Como podemos comprobar, ha desaparecido “Paquete de arroz” pero ha vuelto a aparecer “Leche”.

Esto se debe a que, en el Main, agregamos productos siempre que no existan en el txt, y previamente habíamos eliminado “Leche”, por lo que el programa no ha detectado dicho elemento y lo ha introducido de nuevo en la lista.

5.- Valoración Personal

Me ha parecido muy interesante realizar esta práctica, ya que he podido afianzar los conocimientos adquiridos en la asignatura Acceso a Datos y reforzar los vistos en la asignatura Entornos de Desarrollo del primer año.

Trabajar con programación orientada a objetos me resulta muy atractivo y me motiva a buscar métodos que mejoren el flujo del código y acabe por convertirse en un trabajo con sentido lógico partiendo de una clase vacía.

Siempre es interesante aprender a realizar de manera más optima las tareas que se nos enseñan, y con ello lograr ser cada vez mejor programador.

6.- Bibliografía

Grupo Studium [(↑)](#indice)

1. *María José Martínez Navas, Acceso a Datos (Práctica Tema 1)*. Publicado en **Grupo Studium.**

Recuperado: <https://campustudium.com/pluginfile.php/15072/mod_resource/content/6/Tema%201%20-%20Persistencia%20en%20ficheros%20-%20Pr%C3%A1ctica_v2.pdf>

Último acceso (01/11/2024).