

DAM. UNIDAD 1. ACCESO A FICHEROS PARTE 1. EJERCICIOS NO EVALUABLES

DAM. Acceso a Datos (ADA) (a distancia en inglés)

Unidad 1. ACCESO A LOS FICHEROS

Primera parte. Introducción, repaso de Java y acceso básico a ficheros. **Ejercicios no evaluables**

Abelardo Martínez

Basado y modificado a partir de Sergio Badal
(www.sergiobadal.com) Año 2023-2024

Aspectos a tener en cuenta

Importante

Si buscas las soluciones navegando por Internet o preguntando al oráculo de ChatGPT te estarás engañando a ti mismo. Ten en cuenta que ChatGPT no es infalible ni todopoderoso.

Es una gran herramienta para agilizar tu trabajo una vez que dominas un tema, pero utilizarla como atajo a la hora de adquirir habilidades y conocimientos básicos perjudica seriamente tu aprendizaje. Si lo utilizas para obtener soluciones o consejos por tu cuenta, comprueba también cuidadosamente las soluciones propuestas. Intenta resolver las actividades utilizando los recursos que hemos visto y la documentación ampliada que encontrarás en el "Aula Virtual".

Consejos para programar

Aconsejamos seguir las siguientes normas de codificación:

- Una instrucción por línea.
- Añade comentarios para que tu código sea más claro y legible.
- Utiliza la notación húngara para reconocer el tipo de variables a primera vista.
- Recuerde que hay varias formas de aplicar una solución, así que elija la que más le guste. Recomendamos encarecidamente utilizar `soluciones` basadas en `tampones`.

1. Modo consola. Acceso básico a archivos

Actividad (no evaluable)

Intenta resolver estos 10 ejercicios. No dudes en compartir tus dudas en el foro de UNIT.

1. Escriba un programa Java para obtener una lista de todos los nombres de archivo/directorio del dado.
2. Escriba un programa Java para obtener archivos específicos por extensiones de una carpeta especificada.
3. Escriba un programa Java para comprobar si un archivo o directorio especificado por pathname existe o no.
4. Escriba un programa Java para obtener la última hora de modificación de un fichero.
5. Escriba un programa Java para obtener el tamaño del archivo en bytes, Kb, Mb.
6. Escriba un programa Java para leer el contenido de un archivo línea por línea.
7. Escriba un programa Java para almacenar el contenido de un archivo de texto línea por línea en una matriz.
8. Escriba un programa Java para escribir y leer un archivo de texto plano.
9. Escriba un programa Java para añadir texto a un archivo existente.
10. Escriba un programa Java para encontrar la palabra más larga en un archivo de texto. Elija la forma de implementar: Versión Scanner o BufferedReader.

2. Modo consola. Acceso completo a archivos. Cesta de la compra

Actividad (no evaluable)

Crear un programa en Java para gestionar los PRODUCTOS en un carrito de la compra mediante la impresión y el uso de un menú específico. Después de cada opción, el usuario debe ver el mismo menú hasta que se pulse la opción cero. No dude en compartir sus dudas en el foro de la UNIDAD.

ATENCIÓN: Utilice las excepciones adecuadas cuando acceda a los archivos. Opciones de menú:

- **Pulse 1 para "Preguntar por productos hasta que el usuario introduzca cero como Nombre del producto".**
 - Para cada producto necesitamos el Nombre (String), el Precio (Double) y las Unidades (Integer), añadidos al ArrayList de productos.
 - Una vez introducido el cero como nombre de Producto, todos los productos se guardarán en un archivo "txt" llamado "productos.txt", sobrescribiendo el archivo completo si existe.
 - Se almacena un producto por línea.
 - Después, el menú se imprimirá de nuevo.
- **Pulse 2 para "Listar todos los productos almacenados".**
 - Sólo tienes que leer el archivo "txt" e imprimir cada libro.
- **Pulse 3 para "Eliminar todos los productos".**
 - Sólo tienes que borrar el archivo "txt".
- **Pulse 0 para**

"Salir" Ejemplo de

menú:

```
*****  
Choose an option  
*****  
1. Add Products  
2. List all Products  
3. Remove all Products  
0. Exit  
*****
```

3. Modo gráfico. Acceso completo a archivos. Cesta de la compra

Actividad (no evaluable)

Convierte el programa anterior a un entorno gráfico utilizando la interfaz GUI de Java. No dude en compartir sus dudas en el foro de UNIT.

ATENCIÓN: Utilice las excepciones adecuadas cuando acceda a los archivos.

Recuerde elegir esta versión de Java cuando cree un proyecto GUI:

►  JRE System Library [JavaSE-1.8]

- Cree los objetos gráficos: etiquetas, campos de texto, botones, imágenes. Personaliza los elementos con tu propio diseño.
 - Crear proyecto con máquina Java JRE 1.8.
 - Para distribuir los objetos de forma libre, utilice: JFrame → propiedad Distribución → Distribución absoluta.
 - Por ejemplo:

SHOPPING CART

Product Price Units

SHOPPING CART

Add product

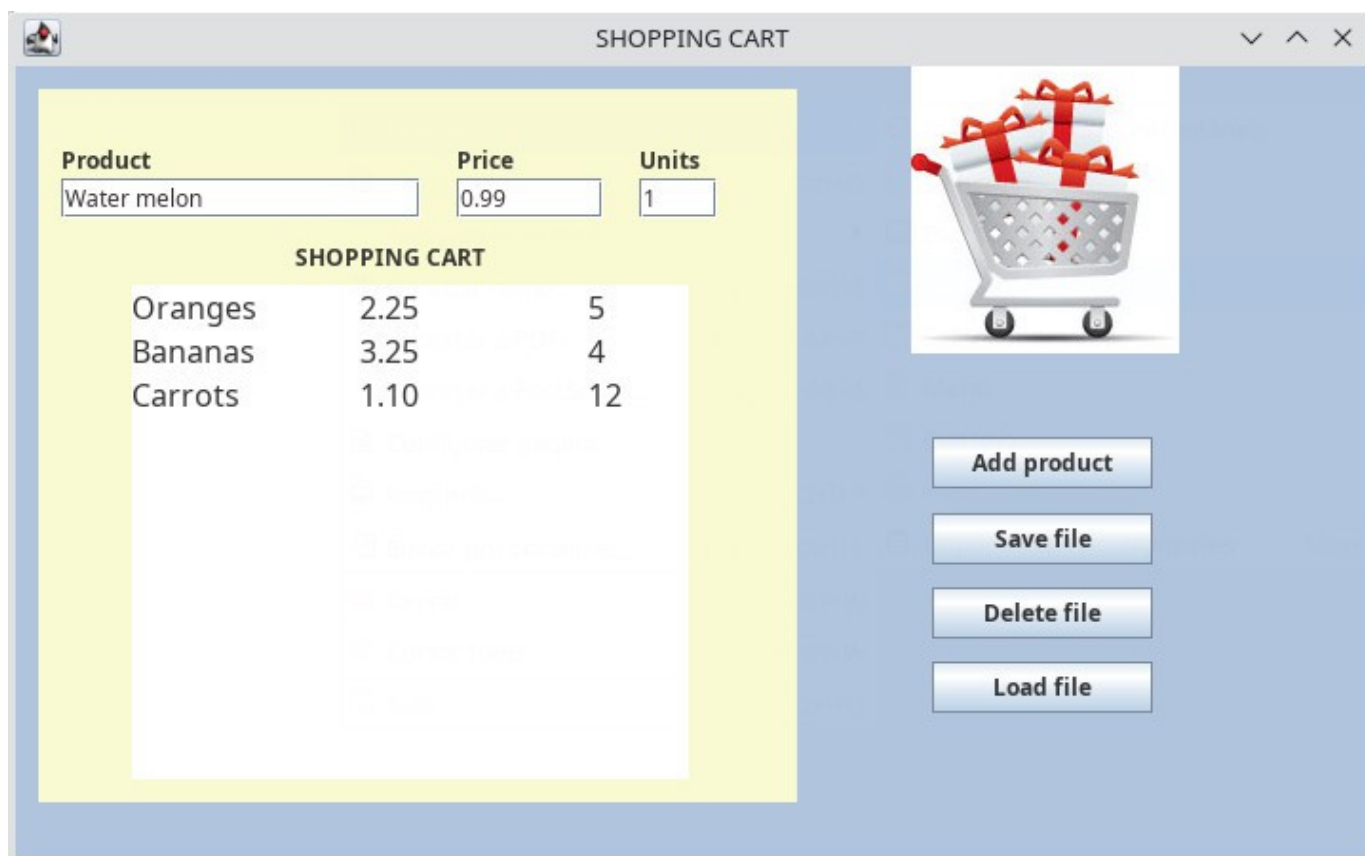
Save file

Delete file

Load file

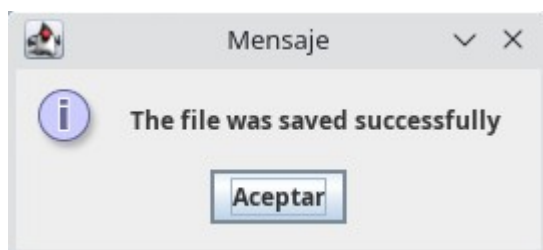
- **Botón Añadir producto.**

- Añade la información del producto (nombre -string-, precio -double- y unidades -integer-) al carrito de la compra (textArea).
- Cada campo debe ir separado por una tabulación.
- Cada producto en una línea diferente.
- Por ejemplo:



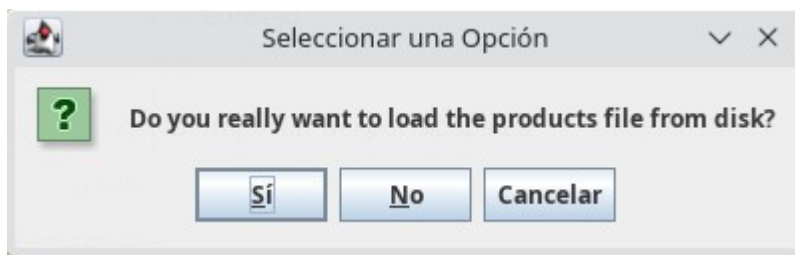
- Botón Guardar archivo.

- Todos los productos se guardarán en un archivo de texto llamado "productos.txt", sobrescribiendo el archivo completo si existe.
- Cada producto en una línea diferente.
- Mostrar un diálogo de mensaje con el resultado. Por ejemplo:



- Botón Cargar archivo.

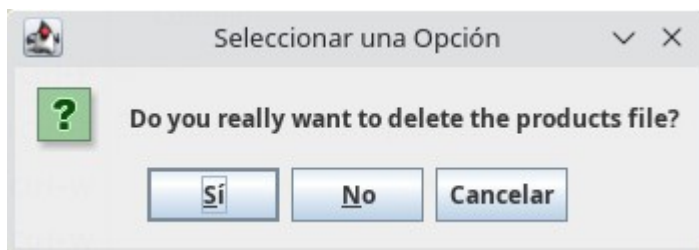
- Pregunte al usuario si está seguro de hacerlo. Por ejemplo:



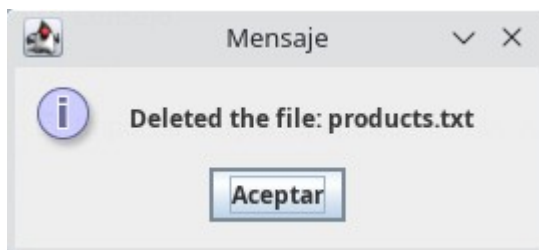
- En caso afirmativo, borre el área de texto, lea el archivo de texto línea por línea y rellene el área de texto con cada producto.

- **Botón Eliminar archivo.**

- Pregunte al usuario si está seguro de hacerlo. Por ejemplo:



- En caso afirmativo, elimine el archivo de texto. Por ejemplo:



- **Salir del programa/programa.**

- Salir por defecto; es decir, al pulsar sobre la "x" se cierra la aplicación.



Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](https://creativecommons.org/licenses/by-sa/4.0/)