

## ACCESO A DATOS – ACTIVIDAD 1: MANEJO DE FICHEROS

Para la realización de la práctica hemos dividido el ejercicio en ramas y dentro de cada rama cada integrante distinto aporte:

Dividimos los requisitos entre los 4 integrantes.

**Néstor:** Primer programa funcional incorporando los elementos básicos y definición del mismo

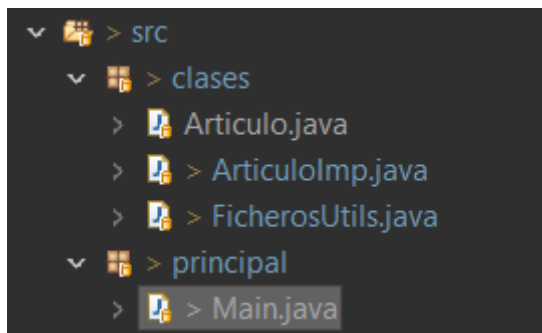
**David:** Incorporación de funciones para operar con artículos, consulta, búsqueda..

**Luis Daniel:** Organización, depuración y debug del programa, control Github.

**Roberto:** Documentación y modularización del programa

En una primera versión se diseña el programa en la misma clase Main apoyándose de una clase Artículo POJO.

Una vez desarrolladas las funciones solicitadas en la rúbrica se modulariza de tal forma que se generan varias clases y se divide según funcionalidades:



**Artículo (POJO):** Incorporará los métodos Getter/setter y un constructor definido.

```
public class Articulo implements Serializable{
    /**
     *
     */
    private static final long serialVersionUID = 1L;
    private int id;
    private String nombre;
    private String descripcion;
    private int cantidad;
    private double precio;

    public Articulo(){
    }

    public Articulo(int id, String nombre, String descripcion, int cantidad, double precio) {
        super();
        this.id = id;
        this.nombre = nombre;
        this.descripcion = descripcion;
        this.cantidad = cantidad;
        this.precio = precio;
    }

    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getNombre() {
        return nombre;
    }
}
```

**ArticuloImp** (Implementación de los métodos propios de artículo, borrar, añadir, consultar...)

```
1 package clases;
2
3 import java.util.ArrayList;
4
5 public class ArticuloImp extends Articulo {
6     /**
7      *
8      */
9     private static final long serialVersionUID = 5298713553698672217L;
10
11     public static void listarArticulos(ArrayList<Articulo> art) {
12         for (Articulo c : art) {
13             System.out.println(c);
14         }
15     }
16
17     public static void anadirArticulo(ArrayList<Articulo> art, Scanner sc) {}
18
19     public static boolean borrarArticulo(ArrayList<Articulo> art, Scanner sc) {
20         System.out.println("Escribe un id");
21         int idBorrar = sc.nextInt();
22         for (Articulo c : art) {
23             if (c.getId() == idBorrar) {
24                 System.out.println(c);
25                 art.remove(c);
26                 return true;
27             }
28         }
29         return false;
30     }
31
32     public static boolean consultarArticulo(ArrayList<Articulo> art, Scanner sc) {}
33 }
```

**FicherosUtils** (realización de las operaciones relativas a ficheros, lectura/escritura).

```
1 package clases;
2
3 import java.io.File;
4
5
6
7
8
9
10
11
12 public class FicherosUtils {
13     private static int id = 0;
14
15     public static void lecturaFichero(ArrayList<Articulo> articulos) {}
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69 }
70
```

La ejecución del programa se realizará en la clase Main.

En primer lugar, se cargará el fichero de datos correspondiente guardado en la raíz del proyecto, mediante la llamada a la clase y método correspondiente.

Se dispone de un menú de selección para poder operar por consola.

```
1
2
3
4 public class Main {
5     static ArrayList<Articulo> articulos = new ArrayList<Articulo>();
6     private static Scanner sc = new Scanner(System.in);
7
8     public static void main(String[] args) throws InterruptedException {
9         System.out.println("CARGANDO FICHERO.... -----");
10        System.out.println("*****");
11        // PRIMERO LEEMOS EL FICHERO Y GENERAMOS EN CASO DE NO EXISTIR
12        FicherosUtils.lecturaFichero(articulos);
13
14        int opcion;
15        do {
16            menu();
17            opcion = sc.nextInt();
18            sc.nextLine();
19
20            switch (opcion) {
21                case 1:
22                    ArticuloImp.anadirArticulo(articulos, sc);
23                    TimeUnit.SECONDS.sleep(2);
24                    break;
25                case 2:
26                    if (ArticuloImp.borrarArticulo(articulos, sc)) {
27                        System.out.println("El artículo a sido eliminado con éxito");
28                    } else {
29                        System.out.println("El id proporcionado no a sido encontrado");
30                    }
31                    TimeUnit.SECONDS.sleep(2);
32                    break;
33            }
34        } while (opcion != 0);
35    }
36 }
```

PRUEBAS DE FUNCIONAMIENTO:

Cargar y Añadir

```
Main [Java Application] C:\Users\rober\p2\pool\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64_17.0.9.v20231028-0858\jre\bin\javaw.exe (28)
CARGANDO FICHERO..... -----
*****
Articulo: Artículo [id=1, nombre=, descripcion=test1, cantidad=5, precio=6.0], A sido añadido con éxito
Articulo: Artículo [id=2, nombre=test1, descripcion=testdescri, cantidad=4, precio=6.0], A sido añadido con éxito
Elige una opción:
Menu:
1 - Añadir nuevo artículo
2 - Borrar artículo por ID
3 - Consulta artículo por ID
4 - Listado de todos los artículos
5 - Terminar el programa
6 - Exportar datos a CSV
1
Escribe un nombre: AGREGADO
Escribe una descripción: AGREGADO
Escribe una cantidad: 22
Escribe un precio: 22
Artículo creado.
Elige una opción:
```

Borrar articulo

```
Articulo [id=1, nombre=, descripcion=test1, cantidad=5, precio=6.0]
Articulo [id=2, nombre=test1, descripcion=testdescri, cantidad=4, precio=6.0]
Articulo [id=3, nombre=AGREGADO, descripcion=AGREGADO, cantidad=22, precio=22.0]
Elige una opción:
Menu:
1 - Añadir nuevo artículo
2 - Borrar artículo por ID
3 - Consulta artículo por ID
4 - Listado de todos los artículos
5 - Terminar el programa
6 - Exportar datos a CSV
2
Escribe un id
1
Articulo [id=1, nombre=, descripcion=test1, cantidad=5, precio=6.0]
El artículo a sido eliminado con éxito
```

Consultar/Listar artículos y Exportar a CSV

Nombre	Fecha de modificación	Tipo	Tamaño
.settings	27/01/2024 13:03	Carpeta de archivos	
bin	27/01/2024 13:03	Carpeta de archivos	
src	27/01/2024 13:03	Carpeta de archivos	
target	27/01/2024 13:04	Carpeta de archivos	
.classpath	26/01/2024 19:15	Archivo CLASSPATH	1 KB
.gitignore	26/01/2024 19:15	Archivo de origen Git Ignore	1 KB
.project	26/01/2024 19:15	Archivo PROJECT	1 KB
articulos	28/01/2024 12:11	Archivo de origen Comma Separated Values	1 KB
articulos.dat	28/01/2024 11:34	Archivo DAT	1 KB
pom	26/01/2024 19:15	Microsoft Edge HTML Document	1 KB

```
Articulo [id=2, nombre=test1, descripcion=testdescri, cantidad=4, precio=6.0]
Articulo [id=3, nombre=AGREGADO, descripcion=AGREGADO, cantidad=22, precio=22.0]
Elige una opción:
Menu:
1 - Añadir nuevo artículo
2 - Borrar artículo por ID
3 - Consulta artículo por ID
4 - Listado de todos los artículos
5 - Terminar el programa
6 - Exportar datos a CSV
6
Artículos exportados a 'articulos.csv' correctamente.
Elige una opción:
Menu:
1 - Añadir nuevo artículo
2 - Borrar artículo por ID
3 - Consulta artículo por ID
4 - Listado de todos los artículos
5 - Terminar el programa
6 - Exportar datos a CSV
```