



Galería de arte

Práctica final 2022-23

Programación III – Grado en
Ingeniería Informática

Diego De Castro Merillas
71043686E

Al principio del programa se comienza en el main en el que iniciamos el método “runMenu” de la clase vista donde se irá ya entrando en los submenús y métodos del programa de la galería de arte.
Este es el método main, donde comienza el programa.

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
4   */
5   package practicafinal2022;
6   import view.View;
7   import com.coti.tools.DiaUtil;
8   /**
9    *
10   * @author ddecem
11   */
12   //Flatlaf dark
13   public class PracticaFinal2022 {
14
15       /**
16        * @param args the command line arguments
17        */
18       public static void main(String[] args) {
19           // TODO code application logic here
20           View v = new View();
21           DiaUtil.startTimerMS();
22           //Menu principal del programa
23
24           v.runMenu("\n 1.- Archivos"
25                   + "\n 2.- Gestion de la Galeria"
26                   + "\n 3.- Listados"
27                   + "\n 4.- Salir\n");
28       }
29   }
```

El programa sigue en la clase View concretamente en el método “runMenu()” que es el menú donde se escogen las opciones de Archivos, gestionar galería, listado de figuras o salir.

```
19   public class View {
20
21       Controller c = new Controller();
22
23       //Metodo del menu principal
24       public void runMenu(String menu){
25           String opcion;
26           boolean salir = false;
27           int n=0;
28
29           if(n==0){
30               c.comienzoPrograma();
31               n++;
32           }
33           System.out.printf(format: "%n\n");
34           Esdia.underline2(prompt: "Menu Principal");
35           do{
36               opcion = Esdia.readString(prompt: menu);
37
38               switch(opcion){
39
40                   case "1" -> {
41                       String menuArchivos = "\na.- Importar CSV\n"
42                                           + "b.- Exportar CSV\n"
43                                           + "c.- Exportar HTML\n"
44                                           + "d.- Volver al menu principal\n"
45                                           + "e.- ";
46                       this.archivosLeerMenu(menuArchivos);
47                   }
48
49                   case "2" -> {
50                       String menuGestionDeGaleria = "\na.- Añadir una figura al inventario\n"
51                                           + "b.- Consultar datos de una figura\n"
52                                           + "c.- Modificar datos de una figura\n"
53                                           + "d.- Eliminar una figura del inventario\n"
54                                           + "e.- Volver al menu principal\n"
55                                           + "\n -> ";
56                       this.gestionDeLaGaleriaLeerMenu(menuGestionDeGaleria);
57                   }
58
59                   case "3" -> {
60                       String listadoMenu = "\na.- Listado por identificador\n"
61                                           + "b.- Listado por año e identificador\n"
62                                           + "c.- Listado por fabricante y año\n"
63                                           + "d.- Volver al menu principal\n"
64                                           + "\n -> ";
65                       this.ListadosLeerMenu(menuListados: listadoMenu);
66                   }
67
68                   case "4" -> {
69                       salir = Esdia.siOno(prompt: "\n ¿Quiere salir del programa? s/n");
70                       if(salir == true){
71                           System.out.printf(format: "%nHa salido del programa\n");
72                           DiaUtil.showFinalTime();
73                           c.salirPrograma();
74                       }
75                   }
76               }
77           }while(salir == false);
78       }
79   }
```

Capturas de todas las bibliotecas importadas en cada archivo

```
package view;
import model.Figura;
import controller.Controller;
import com.coti.tools.Esdia;
import com.coti.tools.DiaUtil;
import com.coti.tools.OpMat;
import java.util.ArrayList;
import java.util.List;
```

```
package controller;
import java.util.List;
import model.Model;
import model.Figura;
/**
```

```
5 package model;
6
7 import java.io.File;
8 import static java.lang.System.out;
9 import java.util.ArrayList;
10 import com.coti.tools.OpMat;
11 import com.coti.tools.Rutas;
12 import com.coti.tools.Esdia;
13 import java.io.BufferedInputStream;
14 import java.io.BufferedOutputStream;
15 import java.io.FileInputStream;
16 import java.io.FileNotFoundException;
17 import java.io.FileOutputStream;
18 import java.io.IOException;
19 import java.io.ObjectInputStream;
20 import java.io.ObjectOutputStream;
21 import java.io.PrintWriter;
22 import java.nio.charset.Charset;
23 import java.nio.file.Files;
24 import java.nio.file.Path;
25 import java.nio.file.StandardOpenOption;
26 import java.util.Collections;
27 import java.util.Comparator;
28 import java.util.List;
29 /**
```

```
package model;

import java.io.Serializable;
```

En el archivo figura se importa "java.io.Serializable" para poder importar y exportar un archivo binario correctamente.

Paquete “view”

Para seguir la ejecución del programa tenemos que entrar en submenús derivados del menú principal, en esos menús ya podremos realizar cambios en la galería ya sea gestionar archivos, galería y listados.

Imagen del submenú archivos

```
81 //Metodo para el submenú de leerArchivos
82 public void archivosLeerMenu(String menuArchivos){
83     String opcion;
84     System.out.println();
85     int a=0;
86     System.out.printf(format: "%n%n%n%n");
87     Esdia.underline2(prompt: "Gestion Archivos");
88     do{
89         opcion = Esdia.readString(prompt: menuArchivos).toLowerCase();
90
91         switch(opcion){
92             case "a" :
93                 c.importarArchivosCSV();
94                 System.out.println();
95                 break;
96
97             case "b" :
98                 c.exportarArchivoCSV();
99                 System.out.println();
100                break;
101
102             case "c" :
103                 c.exportarHTML();
104                 System.out.println();
105                 break;
106             case "d":
107                 System.out.printf(format: "Ha salido del menu leer archivos%n");
108                 a=1;
109                 break;
110             default:
111                 System.out.printf(format: "Introduzca una opcion valida");
112         }
113     }while(a!=1);
114 }
115 // Metodo para el menu de la gestion de la galeria
```

Imagen del submenú gestión de Galería

```
116 // Metodo para el menu de la gestion de la galeria
117 public void gestionDeLaGaleriaLeerMenu(String menuGestionDeLaGaleria){
118     String opcion;
119     int a=0;
120     System.out.printf(format: "%n%n%n%n");
121     Esdia.underline2(prompt: "Menu Gestion de la Galeria");
122     do{
123         opcion = Esdia.readString(prompt: menuGestionDeLaGaleria).toLowerCase();
124
125         switch(opcion){
126             case "a":
127                 this.introducirNuevaPieza();
128                 break;
129             case "b" :
130                 this.consultarDatosDeUnaFigura();
131                 break;
132             case "c" :
133                 this.cambiarDatosFigura();
134                 break;
135             case "d" :
136                 this.eliminarDatosFigura();
137                 break;
138             case "e":
139                 System.out.printf(format: "%nHa salido del menu gestion de galeria");
140                 a=1;
141                 break;
142             default:
143                 System.out.printf(format: "%nNo ha introducido la opcion correcta%n");
144         }
145     }while(a == 0);
146 }
```

Imagen del submenú Listados de figuras

```
144 // Metodo del menu de listado de las figuras
145 public void ListadosLeerMenu(String menuListados){
146     String opcion;
147     System.out.printf(format: "%n%n%n%n");
148     Esdia.underline2(prompt: "Listado de las figuras");
149     do{
150         opcion = Esdia.readString(prompt: menuListados).toLowerCase();
151
152         switch(opcion){
153             case "a" :
154                 this.listadoPorIdentificador();
155                 break;
156             case "b" :
157                 this.listadoPorAñoIdentificador();
158                 break;
159             case "c" :
160                 this.listadoPorFabricanteYAño();
161                 break;
162             case "d" :
163                 System.out.printf(format: "Ha salido de la opcion listados%n");
164                 break;
165             default :
166                 System.out.println(s: "No ha introducido ninguna de las posibilidades");
167         }
168     }while(!"d".equals((String) opcion));
169 }
170 }
```


En el submenú archivos tenemos tres opciones, “importarArchivosCSV()”, “ExportarArchivosCSV()” y “ExportarHTML()” que están en la clase model. Pondré las fotos cuando explique la clase Model.

En el submenú gestión de galería hay 4 opciones, “introducirNuevaPieza()”, “consultarDatosDeUnaFigura()”, “cambiarDatosFigura()” y “eliminarDatosFigura()”.

```

171 ////////////////////////////////////////////////// MENU GESTION DE GALERIA ////////////////////////////////////////////
172 //Metodo introducir nueva pieza --> aqui se preguntan los valores y se comprueba si se ponen bien
173 public void introducirNuevaPieza() { .....39 líneas }
174
213 // Metodo Consultar Datos de una figura, se pregunta el identificador y se comprueba si existe y luego se imprime por pantalla
214 public int consultarDatosDeUnaFigura() { .....32 líneas }
215
244 //Metodo cambiar datos
245 // Se pide el identificador de la figura y si existe se pide al usuario que elija que cambiar al usuario
246 public int cambiarDatosFigura() { .....87 líneas }
247
335 //Metodo eliminar datos
336 public int eliminarDatosFigura() { .....36 líneas }
337
372
373

```

Método “introducirNuevaPieza()”

```

373 //Metodo introducir nueva pieza --> aqui se preguntan los valores y se comprueba si se ponen bien
374 public void introducirNuevaPieza() {
375     System.out.printf(Formato: "%n\n\n");
376     int dev;
377     String identifi;
378     boolean comp = true;
379     Eadla.underline2(Formato: "Añadir una figura al inventario");
380     do {
381         identifi = Eadla.readString(Formato: "Introduzca un identificador para la pieza\n-> ");
382         dev = c.piezaBucle(identifi);
383         if(dev == 1) {
384             System.out.println("Ha introducido un identificador ya existente, introduzca otro por favor\n-> ");
385         }
386     } while(dev == 1);
387
388     float alturaMetros = Eadla.readFloat(Formato: "Introduzca la altura en metros\n-> ", (float) 0.001, (float) 1.5); //minimo la figura tiene que tener una altura, ya que biblioteca.jar solo
389                                                         //como valor minimo, pongo un valor minimo para que la figura siempre tenga
390     String material = Eadla.readString(Formato: "Introduzca el material con el que esta construido en su mayoria\n-> ");
391
392     int cantidad;
393     do {
394         cantidad = Eadla.readInt(Formato: "Introduzca la cantidad de figuras de ese modelo\n-> ");
395     } while(cantidad < 1); //Como no tiene un maximo he hecho un bucle solo con el numero
396
397     int ano = Eadla.readInt(Formato: "Introduzca el año de construcción del modelo de la figura\n-> ", ano 0, ano 2100);
398     String foto;
399     do {
400         comp = true;
401         foto = Eadla.readString(Formato: "Introduzca el nombre del archivo .PNG\n-> ");
402         if(false == foto.endsWith(Formato: ".png").toLowerCase()) {
403             comp = false;
404             System.out.printf(Formato: "Introduzca el nombre con .png al final\n");
405         }
406     } while(comp == false);
407     String fabricante = Eadla.readString(Formato: "Indíqueme cual es el fabricante\n-> ");
408     fabricante = c.ponerPrimerLetraMayuscula(Formato: fabricante);
409     //fabricante.toUpperCase().charAt(0) = fabricante.substring(0, fabricante.length()).toLowerCase();
410     c.introducirNuevaPieza(identifi, alturaMetros, material, cantidad, ano, foto, fabricante);
411 }

```

Método “consultarDatosDeUnaFigura()”

```

214 // Metodo Consultar Datos de una figura, se pregunta el identificador y se comprueba si existe y luego se imprime por pantalla
215 public int consultarDatosDeUnaFigura() {
216     System.out.printf(Formato: "%n\n\n");
217     Eadla.underline2(Formato: "Consulta de datos de una figura");
218     int ret;
219     String identificador;
220     List <Figura> fig = new ArrayList<>();
221
222     //Hará el bucle hasta que haya un identificador valido
223     identificador = Eadla.readString(Formato: "Introduzca el identificador de la figura\n-> ");
224     ret = c.piezaBucle(Formato: identificador); //Es un metodo en el que compruebo si el identificador es igual o no
225     if(ret == 2) {
226         System.out.println("Ha introducido un identificador que no existe, se vuelve al menu de gestion");
227         return 0;
228     }
229     else {
230         fig = c.consultarFigura(identificador); //tengo la figura que buscamos
231         //Se imprime los datos de la figura
232         for(Figura f : fig) {
233             System.out.println("La figura buscada es:");
234             //System.out.println(devolverFormato());
235             System.out.printf("%tIdentificador %t\n",
236                             + "%tAltura(M) %t\n",
237                             + "%tMaterial %t\n",
238                             + "%tCantidad %t\n",
239                             + "%tAño %t\n",
240                             + "%tFoto %t\n",
241                             + "%tFabricante %t\n",
242                             + "%t", f.getIdentificador(), + "%t", f.getAlturaMetros(),
243                             + "%t", f.getMaterial(), + "%t", f.getCantidad(), + "%t", f.getAño(), + "%t", f.getFoto(), + "%t", f.getFabricante());
244             return 1;
245         }
246     }
247     //Metodo cambiar datos
248     // Se pide el identificador de la figura y si existe se pide al usuario que elija que cambiar al usuario

```

Método “cambiarDatosFigura()”

```

148 public int cambiarDatosFigura() {
149     System.out.println("Introduzca el identificador de la figura que desea cambiar:");
150     int id;
151     String opcion;
152     boolean o = false;
153     Edia.underline("Cambiar Datos de la figura");
154     String identificador;
155     float alturaM = 0.0f;
156     String material = "xxx";
157     int cantidad = 2;
158     int ano = 0;
159     String foto = "xxx.png";
160     String fabricante = "xxxxx";
161     List<Figura> fig = new ArrayList<>();
162     //Borra el bucle hasta que haya un identificador valido
163     identificador = Edia.readString(prompt: "Introduzca el identificador de la figura-> ");
164     ret = o.piezaBucle(identificador); //Se un metodo en el que compruebo si el identificador es igual o no
165     if(ret == 2){
166         System.out.println("Ha introducido un identificador que no existe, se vuelve al menu de gestion");
167         return 0;
168     }
169     else{
170         fig = o.consultarFigura(identificador); //tengo la figura que buscamos
171         // Introduzco en variables los valores de la figura para asi poder cambiarlos luego
172         for(Figura f : fig){
173             System.out.println("La figura buscada es:");
174             //System.out.println(devolverFormato());
175             System.out.println(prompt: "Selecione la figura a modificar:");
176             id = f.getIdentificador();
177             alturaM = f.getAlturaMetros();
178             material = f.getMaterial();
179             cantidad = f.getCantidad();
180             ano = f.getAno();
181             foto = f.getFoto();
182             fabricante = f.getFabricante();
183         }
184     }
185     // Menu para elegir quedatos cambiar
186     do{
187         opcion = Edia.readString("Introduzca que valor quiere cambiar menos el identificador:\n"
188             + "1- Altura\n"
189             + "2- Material\n"
190             + "3- Cantidad\n"
191             + "4- Año\n"
192             + "5- Foto\n"
193             + "6- Fabricante\n"
194             + "s- Salir-> ");
195     }
196     switch(opcion){
197         case "1":
198             alturaM = Edia.readFloat(prompt: "Introduzca el nuevo valor de la altura-> ", (float) 0.001, (float) 1.5);
199             break;
200         case "2":
201             material = Edia.readString(prompt: "Introduzca el material nuevo-> ");
202             break;
203         case "3":
204             do{
205                 cantidad = Edia.readInt(prompt: "Introduzca el cantidad de figuras del modelo->");
206                 if(cantidad < 1){
207                     System.out.println("Introduzca una cantidad mayor que 0");
208                 }while(cantidad<1);
209             }break;
210         case "4":
211             ano = Edia.readInt(prompt: "Introduzca el nuevo año de construcción de la figura->", 0, 2100);
212             break;
213         case "5":
214             boolean comp=true;
215             do{
216                 foto = Edia.readString(prompt: "Introduzca el nombre del archivo .PNG-> ");
217                 if(!foto.endsWith(".png")){
218                     comp = false;
219                     System.out.println("Introduzca el nombre con .png al final");
220                 }
221             }while(comp == false);
222             break;
223         case "6":
224             fabricante = Edia.readString(prompt: "Introduzca el nuevo fabricante->");
225             break;
226         case "s":
227             //return - true if y, false if n
228             o = Edia.siOnc(prompt: "Quiere salir?");
229             o.cambiarDatosFigura(identificador, alturaM, material, cantidad, ano, foto, fabricante);
230             break;
231     }
232 }while(o == false);
233 }

```

Método “eliminarDatosFigura()”

```

327 public int eliminarDetosFigura() {
328     DiaUtil.clear();
329     System.out.printf(format, "%n%n%n");
330     int opcion;
331
332     List <Figura> fig = new ArrayList<>();
333     Eddia underline(prompt, "Eliminar Figura");
334     String identificador = Eddia.readString(prompt, "Introduzca el identificador de la pieza a eliminar(nº):");
335     ret = c.piezaBuscable(identificador); //Se usa un metodo en el que compruebo si el identificador es igual o no
336     if(ret == 2){
337         System.out.println("La ha introducido un identificador que no existe, se vuelve al menu de gestion");
338         return 0;
339     }
340     else{
341         fig = c.consultarFigura(identificador); //tengo la figura que buscamos
342
343         for(Figura f : fig){
344             System.out.println("La figura buscada es:n");
345             System.out.printf(format, "%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t%s\t", args.f.getIdentificador(), args.f.getAlturaMetros(),
346                 args.f.getMateria(), args.f.getCantidad(), args.f.getAño(), args.f.getFoto(), args.f.getFabricante());
347             //return - true if y, false if n
348             System.out.println(Eddia.readString(prompt, "Desea eliminar la pieza de verdad?n1- Sí 2- No\n-> ", min: 1, max: 2));
349             if(opcion == 1){
350                 c.eliminarPieza(pieza.identificador);
351                 System.out.printf(format, "Se ha eliminado la figura %s", args.identificador);
352             }
353             else{
354                 System.out.printf(format, "No se ha eliminado la figura, se vuelve al menu de gestion");
355             }
356         }
357     }
358     return 0;
359 }

```

Por último, en la clase View tenemos los métodos “listadoPorIdentificador()”, “listadoPorAñoEIdentificador()”, “listadoPorFabricanteYAño()” y “mostrarPorPantalla()”.

Estos métodos son muy parecidos entre sí, contienen varios métodos uno para ordenar, que es cada uno diferente, el primero se ordenará por identificador, el segundo por año descendente e identificador y el tercero por fabricante y año. Estos métodos al ejecutarlos quedasen ordenados para luego escribirlos por pantalla.

El segundo método es “`imprimirConFormato()`” en el que esta en el modelo y se pasa el `ArrayList` a una matriz bidimensional de Strings para poder luego usar la función `printToScreen3`.

El método que usa `printToScreen3` es “`mostrarPorPantalla()`”.

```

375 /////////////////////////////////////////////////// MENU LISTADOS ////////////////////////////////////////
376
377 [x] public void listadoPorIdentificador() [...13 lines]
378
379 [x] public void listadoPorAnoIdentificador() [...11 lines]
380
381 [x] public void listadoPorFabricanteYAno() [...10 lines]
382
383 [x] public void mostrarPorPantalla(String[][] figuras) [...10 lines]
384
385
386 }
387

```

| | | | | |
|--------|------------------------------|--|---------------------------------|--|
| Método | "listadoPorIdentificador()" | | "listadoPorAnoEidentificador()" | |
| | "listadoPorFabricanteYAno()" | | | |

```

377 public void listadoPorIdentificador(){
378     System.out.printf(format: "%n%n%n%n%n");
379     //Se ordenan los valores
380     c.listadoPorIdentificadorC();
381
382     String[][] listado = c.imprimirConFormato();
383
384
385     System.out.printf(format: "El listado de las figuras ordenadas por identificador es%n");
386     mostrarPorPantalla(figuras: listado);
387
388 }
389
390
391 public void listadoPorAñoIdentificador(){
392     System.out.printf(format: "%n%n%n%n%n");
393     //Ordenamos los valores por año e identificador si tienen el mismo año
394     c.listadoPorAñoIdentificador();
395
396     String[][] listado = c.imprimirConFormato();
397
398     System.out.printf(format: "El listado de las figuras ordenadas por año e identificador es%n");
399     mostrarPorPantalla(figuras: listado);
400
401 }
402
403
404 public void listadoPorFabricanteYano(){
405     System.out.printf(format: "%n%n%n%n%n");
406
407     c.listadoPorFabricanteYano();
408
409     String[][] listado = c.imprimirConFormato();
410
411     System.out.printf(format: "El listado de las figuras ordenadas por fabricante y año es%n");
412     mostrarPorPantalla(figuras: listado);
413
414 }

```

Método “mostrarPorPantalla()”

```

415 public void mostrarPorPantalla(String[][] figuras){
416     System.out.printf(formas: "%1s%1s%1s%1s%1s%1s%1s", args: "Identific",
417                                     // Hacemos la función donde se comprueba printToScreen3 para así reciclar
418                                     args: "Altura (M)", args: "Material", args: "H", args: "Año", args: "Foto", args: "Fabricante");
419     try{
420         OpMat.printToScreen3(matriz: figuras);
421     }catch(Exception ex){
422         System.out.printf(formas: "No fue posible mostrar los valores que quiere imprimir con el metodo printToScreen3");
423     }
424 }

```

Paquete “controller”

En la clase Controlador se pasan los métodos de la vista a el modelo.

```
12 //
13 public class Controller {
14     Model m = new Model();
15
16     ////////////////////////////////////////////////// METODOS PARA PASAR DEL MODELO A LA VISTA Y AL REVES ///////////////////////////////////
17
18     public void comienzoPrograma() {
19         m.comienzo();
20     }
21
22
23     public void salirPrograma() {
24         m.finalPrograma();
25     }
26
27
28     public void importarArchivosCSV() {
29         m.importarCSV();
30     }
31
32     public void exportarArchivoCSV() {
33         m.exportarFigurasCSV();
34     }
35
36
37     public void introducirNuevaPieza(String identifi, float alturaMetros, String material, int cantidad, int ano, String foto, String fabricante) {
38         m.introducirNuevaPiezas(identifi, alturaMetros, material, cantidad, ano, foto, fabricante);
39     }
40
41     public void eliminarPieza(String pieza) {
42         m.eliminarPieza(pieza);
43     }
44
45     public List<Figura> consultarFigura(String identificador) {
46         return m.consultarFiguraId(identificador);
47     }
48
49     public void cambiarDatosFigura(String identificador, float alturaM, String material, int cantidad, int ano, String foto, String fabricante) {
50         m.cambiarDatos(identificador, alturaM, material, cantidad, ano, foto, fabricante);
51     }
52
53
54     public int piezaBucle(String identifi) {
55         int dev = m.piezaBucle(identifi);
56         return dev;
57     }
58
59     public void exportarHTML() {
60         m.exportarFigurasHTML();
61     }
62
63     public void listadoPorIdentificadorC() {
64         m.listadoPorIdentificador();
65     }
66
67     public void listadoPorAnoEIdentificador() {
68         m.listadoPorAnoEIdentificador();
69     }
70
71     public void listadoPorFabricanteYAno() {
72         m.listadoPorFabricanteYAno();
73     }
74
75     public void print(List<Figura> f) {
76         m.printPantalla(f);
77     }
78
79     public String[][] imprimirConFormato() {
80         return m.imprimirConFormato();
81     }
82
83     public String ponerPrimeraLetraMayuscula(String str) {
84         return m.ponerPrimeraLetraMayuscula(str);
85     }
86
87 }
```


Paquete “model” → Clase Figura

```
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4   */
5  package model;
6
7  import java.io.Serializable;
8
9  /**
10   *
11   * @author Gecm
12   */
13  public class Figura implements Serializable{
14
15      //Atributos de la clase Figura
16      private String identificador; //Identificador de la figura --> No hay otro identificador igual
17      private float alturaMetros; // Dimensiones de la figura en metros
18      private String material; // Material de construcción mayoritario de la figura
19      private int cantidad; // el número de figuras de un modelo que tiene la galería
20      private int ano; // Año en el que se compró la figura
21      private String foto; // Nombre del archivo gráfico de tipo png
22      private String fabricante; // nombre del creador de la figura
23
24      //Constructor sin paso de parametros
25      public Figura(){
26          this.identificador = "1000000";
27          this.alturaMetros = (float) 0.01;
28          this.material = "xxxxx";
29          this.cantidad = 1;
30          this.ano = 2000;
31          this.foto = "aroh.png";
32          this.fabricante = "xxxxxxx";
33      }
34
35      /**
36       *
37       * @param identif
38       * @param alturaMet
39       * @param material
40       * @param cantidad
41       * @param ano
42       * @param foto
43       * @param fabricante
44       */
45      // Constructor con paso de parametros que se le asigna unos valores a la figura
46      public Figura(String identif , float alturaMet , String material , int cantidad , int ano , String foto , String fabricante ){
47          this.identificador = identif;
48          this.alturaMetros = alturaMet;
49          this.material = material;
50          this.cantidad = cantidad;
51          this.ano = ano;
52          this.foto = foto;
53          this.fabricante = fabricante;
54      }
55
56      // Metodo para tratar la exportación de ficheros, cada celda de la fila tiene un tipo atomico que hay que ver si la informacion del
57      // archivo se puede convertir al que es, es decir tiene el mismo tipo atomico.
58      //Se importa en formato String
59      public static Figura factory(String[] tmp){
60          float tmp1;
61          int tmp3,tmp4;
62          for(String s : tmp){
63              if(s.isBlank()){
64                  return null;
65              }
66              //retorna null si alguna de las opciones de la figura esta a NULL, ya que no debe haber ninguna
67          }
68
69          //Hemos importado un array de String, eso significa que tendremos que pasar todos los valores a los tipos atomicos que necesitamos
70          if(tmp[0] == null){
71              return null;
72          }
73          //Hacemos comprobacion si la celda 5 que es foto, contiene .png al final ya que es necesario
74          /*
75          if(!tmp[5].endsWith(".png")){
76              return null;
77          }
78          */
79          //Cambiamos a Float o int dependiendo la variable
80          try {
81              tmp1 = Float.parseFloat(tmp[1]);
82              tmp3 = Integer.parseInt(tmp[3]);
83              tmp4 = Integer.parseInt(tmp[4]);
84          } catch (NumberFormatException e) {
85              //Si el valor de la altura se retorna null significa que en la posicion de la altura no estaba un valor correcto
86              return null;
87          }
88          // Creamos la figura llamando al constructor y la retornamos al metodo que habia llamado a este.
89          return new Figura(tmp[0], alturaMet=tmp1, tmp[2], cantidad=tmp3, ano=tmp4, tmp[5], tmp[6]);
90      }
91
92      //Metodo que devuelve en formato String la informacion sobre una figura
93      public String escribirFicheroConDelimiter(String DELIMITER){
94          return String.format("%s%s%s%s%s%s%s%s",
95              this.identificador, args= DELIMITER,
96              this.alturaMetros, args= DELIMITER,
97              this.material, args= DELIMITER,
98              this.cantidad, args= DELIMITER,
99              this.ano, args= DELIMITER,
100              this.foto, args= DELIMITER,
101              this.fabricante);
102      }
103
104      public String[] valoresFigura(){
105          String[] tmp = new String[7];
106          tmp[0] = this.identificador;
107          tmp[1] = String.valueOf(this.alturaMetros);
108          tmp[2] = this.material;
109          tmp[3] = String.valueOf(this.cantidad);
110          tmp[4] = String.valueOf(this.ano);
111          tmp[5] = this.foto;
112          tmp[6] = this.fabricante;
113
114          return tmp;
115      }
116  }
```

Como podemos ver en las primeras líneas declaramos los atributos de la clase figura que son 7, indicados en el código cada uno su nombre.

Posteriormente tenemos dos constructores, uno para cuando se crea una Figura sin datos y otra cuando creamos una figura con unos datos determinados.

En este fichero también nos encontramos con el método “factory” que lo usamos para poder pasar del fichero CSV a un ArrayList, donde comprobamos si los datos están en el formato correcto y con ellos creamos una figura que luego añadimos a el ArrayList.

Los dos siguientes métodos, son usados para imprimir con formato, el primero con el delimitador tabulador y el segundo lo usamos para pasar los valores a una matriz de Strings con los que podremos escribir los valores con “println3”.

Getters y Setters de los atributos de la clase figura

Con estos atributos podremos coger los datos cuando los necesitamos y cambiarlos.

```
135 ////////////////////////////////////////////////// GETTERS Y SETTERS ////////////////////////////////////////////
136 public String getIdentificador() {
137     return identificador;
138 }
139 public void setIdentificador(String identificador) {
140     this.identificador = identificador;
141 }
142
143 public float getAlturaMetros() {
144     return alturaMetros;
145 }
146 public void setAlturaMetros(float alturaMetros) {
147     this.alturaMetros = alturaMetros;
148 }
149
150 public String getMaterial() {
151     return material;
152 }
153 public void setMaterial(String material) {
154     this.material = material;
155 }
156
157 public int getCantidad() {
158     return cantidad;
159 }
160 public void setCantidad(int cantidad) {
161     this.cantidad = cantidad;
162 }
163
164 public int getAño() {
165     return año;
166 }
167 public void setAño(int año) {
168     this.año = año;
169 }
170
171 public String getFoto() {
172     return foto;
173 }
174 public void setFoto(String foto) {
175     this.foto = foto;
176 }
177
178 public String getFabricante() {
179     return fabricante;
180 }
181 public void setFabricante(String fabricante) {
182     this.fabricante = fabricante;
183 }
```

Por último, nos encontramos la clase “model” donde esta hecho toda la lógica de cambiar, meter los datos en el ArrayList y variables.

```
//
public class Model {

    private final String NAME_FILE_CSV = "figuras.csv"; //Nombre del fichero .csv
    private final String NAME_FOLDER = "datos_figuras"; //Nombre de la carpeta donde se encuentra los ficheros .bin .csv y el HTML
    private final String NAME_HTML = "figuras.html"; //Nombre de la pagina HTML
    private final String NAME_FILE_BIN = "figuras.bin"; //Nombre del fichero .bin
    private List<Figura> generalList = new ArrayList<>();
    private final int CAMPO = 7; //Numero de campos de la figura | Identificacion | Altura | Material | Cantidad | Año | Foto | Fabricante
```

Declaramos al principio las variables de los nombres de los archivos y la carpeta que los contiene junto con el ArrayList donde meteremos todas las figuras y la variable campos.

Método “comienzo()”

Este método contiene el código relativo a la importación de archivo binario y si no un archivo CSV.

```
44 public void comienzo(){
45
46     // Guardamos la direccion donde esta el fichero
47     Path ficheroBin = Rutas.pathToFileInFolderOnDesktop( nameOfFolder:NAME_FOLDER, nameOfFile:NAME_FILE_BIN);
48
49     //Si existe el fichero entonces importamos la informacion de comienzo del fichero binario
50     if(ficheroBin.toFile().exists()){
51
52         FileInputStream fis;
53         BufferedInputStream bis;
54         ObjectInputStream ois;
55
56         try{
57             //Declaro estos metodos para asi importar el fichero
58             fis = new FileInputStream(ficheroBin.toFile());
59             bis = new BufferedInputStream(fis);
60             ois = new ObjectInputStream(bis);
61
62             //Guardamos en generalList la informacion del fichero binario
63             generalList = (List<Figura>) ois.readObject();
64             System.out.printf(format:"\n\nSe ha leído el archivo binario correctamente\n\n");
65             ois.close();
66         }catch (IOException | ClassNotFoundException ex){
67             System.out.printf(format:"No se ha podido leer el archivo\n\n");
68             System.out.printf(format:"Asegurese que la carpeta datos_figuras esta en el escritorio y el archivo figuras.bin esta dentro\n\n\n\n");
69         }
70
71         //Si no existe o no se puede importar el fichero binario entonces como segunda opcion importamos un fichero CSV que ayuda a empezar
72         else{
73             System.out.printf(format:"No existe el fichero\n\n");
74             System.out.printf(format:"\n\nNo se ha leído el fichero binario por lo que leamos el CSV\n\n");
75             System.out.printf(format:"Se importa un fichero CSV para poder comenzar el programa\n\n");
76             this.importarCSV();
77         }
78
79         // Metodo que se llama al final del programa para asi exportar la informacion actualizada de la sesion en un fichero binario
80     public void finalPrograma(){
81         Path ficheroBin = Rutas.pathToFileInFolderOnDesktop( nameOfFolder:NAME_FOLDER, nameOfFile:NAME_FILE_BIN);
82         try{
83             FileOutputStream fos = new FileOutputStream(ficheroBin.toFile());
84             BufferedOutputStream bos = new BufferedOutputStream(fos);
85             ObjectOutputStream oos = new ObjectOutputStream(bos);
86
87             //Escribe la informacion de la lista generalList en el fichero CSV
88             oos.writeObject(generalList);
89             oos.close();
90         }catch(IOException ex){
91             System.out.printf(format:"No fue posible guardar el archivo binario con las figuras");
92         }
93     }
```

El método “finalPrograma()” exporta un archivo binario.

```
78 // Metodo que se llama al final del programa para asi exportar la informacion actualizada de la sesion en un fichero binario
79 public void finalPrograma(){
80     Path ficheroBin = Rutas.pathToFileInFolderOnDesktop( nameOfFolder:NAME_FOLDER, nameOfFile:NAME_FILE_BIN);
81     try{
82         FileOutputStream fos = new FileOutputStream(ficheroBin.toFile());
83         BufferedOutputStream bos = new BufferedOutputStream(fos);
84         ObjectOutputStream oos = new ObjectOutputStream(bos);
85
86         //Escribe la informacion de la lista generalList en el fichero CSV
87         oos.writeObject(generalList);
88         oos.close();
89     }catch(IOException ex){
90         System.out.printf(format:"No fue posible guardar el archivo binario con las figuras");
91     }
92 }
```

Método “importarCSV()”

```
98 public int importarCSV(){
99
100     final String DELIMITADOR = "\t";
101     int n = 0, a=0;
102     Eadia.underLine2(pagep: "Método: importarCSV()");
103
104     //Cogemos la ruta donde est el fichero CSV
105     File archivoCSV = Rutas.pathToFileInFolderOnDesktop(nameOfFolder: NAME_FOLDER, nameOfFile: NAME_FILE_CSV).toFile();
106     System.out.println("La ruta donde están los ficheros es " + archivoCSV);
107
108     //Si el archivo existe intentamos iportar la informacion
109     if (archivoCSV.exists()) {
110         //tmp guarda la informacion del fichero csv
111         String[] tmp;
112         try {
113             tmp = OpMat.importFromDisk(f: archivoCSV, delimiter: DELIMITADOR);
114         } catch (Exception ex) {
115             out.printf(format: "%nNo fue posible importar el archivo");
116             return 0;
117         }
118         //Iteramos cada fila de tmp para ver si podemos pasarlo a ArrayList y los datos estan en un formato bueno
119         generalList = new ArrayList<>();
120         for (String[] campos : tmp) { //for(int i=0; i<tmp[i].length; i++)
121             Figura f;
122             f = Figura.factory(seg: campos);
123             if (null != f) {
124                 generalList.add(f);
125                 a = a + 1;
126             } else {
127                 System.out.printf(format: "Se ha descartado la línea %d %n", a+1);
128                 n = n + 1;
129             }
130         }
131     }
132     else{
133         System.out.printf(format: "No existe el archivo");
134     }
135     //Informacion de los errores que ha dado el fichero
136     if(generalList != null){
137         System.out.printf(format: "%nSe ha leído correctamente el fichero%n");
138         int total = n+a;
139         float percent = (total*n)/a;
140         System.out.printf("El número de registros descartados es "+ n + " de un total de " + total + " y su porcentaje de errores es de %.2f%% %n", seg: percent );
141     }
142     return 1;
143 }
```

Método “ExportarCSV()”

```
144
145 public void exportarFigurasCSV(){
146     //ruta del fichero dnde se guardará la informacion
147     final String DELIMITER = "\t";
148     File exportarFiguras = Rutas.pathToFileInFolderOnDesktop(nameOfFolder: NAME_FOLDER, nameOfFile: "figuras.csv").toFile();
149
150     if(null == generalList){
151         System.out.printf(format: "La lista de figuras esta vacia, no puede exportar ningun elemento");
152     }
153
154     List<String> tmp = new ArrayList<>();
155     for(Figura f : generalList ){
156         tmp.add(f.f.escrIBEElFicheroConDELIMITER(DELIMITER));
157     }
158
159     try{
160         //Escribe en el fichero en formato csv de la informacion contenia en la lista tmp
161         Files.write(path: exportarFiguras.toPath(), lines: tmp, charset: Charsets.defaultCharset(), options: StandardOpenOption.CREATE);
162     }catch(IOException ex){
163         System.out.printf(format: "No se pudo crear el archivo exportado");
164     }
165     System.out.printf(format: "El archivo se ha exportado correctamente");
166 }
```

Método “ExportarHTML()”

```
167 // Metodo que exporta a HTML la lista con las figuras
168 public void exportarFigurasHTML(){
169     File exportarHTML = Rutas.pathToFileInFolderOnDesktop(nameOfFolder: NAME_FOLDER, nameOfFile: NAME_HTML).toFile();
170     if(exportarHTML.exists()){
171         exportarHTML.delete();
172     }
173     try{
174         //Creamos un PrintWriter para escribir en el fichero
175         PrintWriter pw = new PrintWriter(file: exportarHTML);
176         //Formato de como va a ser el encabezado de la tabla
177         pw.printf("<DOCTYPE html>%n"
178             + "<html>%n"
179             + "<head>%n"
180             + "<meta charset='UTF-8'>%n"
181             + "</head>%n"
182             + "<body>%n");
183         pw.printf(format: "<table border=1>%n");
184         pw.printf(format: "<caption>GALERIA DE ARTE</caption>"); //Titulo de la tabla
185         pw.printf("<tr><tr style='background-color: rgb(102, 255, 153);>%n" //Titulo de cada columna
186             + "<td>IDENTIFICADOR</td>" //IDENTIFICACION
187             + "<td>ALTURA (M)</td>" //ALTURA M
188             + "<td>MATERIAL</td>" //MATERIAL
189             + "<td>CANTIDAD</td>" //CANTIDAD
190             + "<td>AÑO</td>" // AÑO
191             + "<td>FOTO</td>" //FOTOS
192             + "<td>FABRICANTE</td>" //FABRICANTES
193             + "</tr>");
194         // Con este bucle imprimimos todos los valores en el fichero HTML
195         for(Figura f : generalList){
196             pw.printf(format: "%s%n", seg: f.asHTML());
197         }
198         pw.printf(format: "</tr></body>%n</html>%n");
199         pw.close();
200     }catch(FileNotFoundException ex){
201         System.out.println("ERROR AL EXPORTAR A HTML");
202     }
203     System.out.printf(format: "Se ha exportado el fichero HTML correctamente");
204 }
```

Método “piezaBucle()” “introducirNuevaPieza()” “consultarFiguraId()”

```
208 // Metodo para ver si esta la figura en el
209 public int piezaBucle(String identificadorNew){
210     String identif;
211     String newId = identificadorNew;
212     for(Figura f : generalList){
213         identif = f.getIdentificador();
214         if(newId.equalsIgnoreCase(identif)){
215             //Significa que hay un identificador igual
216             return 1;
217         }
218     }
219     /*
220     if(true == generalList.contains(identificadorNew)){
221         return 1;
222     }*/
223     return 2;
224 }
225 //Metodo introducir nueva pieza en ArrayList
226 public void introducirNuevaPieza(String identif, float alturaMetros, String material, int cantidad, int ano, String foto, String fabricante){
227
228     Figura f = new Figura(identif, alturaMetros.alturaMetros, material, cantidad, ano, foto, fabricante);
229     generalList.add(f);
230     System.out.printf(format, "Se ha añadido la figura al inventario\n");
231 }
232 //Metodo consultarFigura por Identificador
233 public List<Figura> consultarFiguraId(String identificador){
234     String valor;
235     List<Figura> figura = new ArrayList<>();
236     for(Figura f : generalList){
237         valor = f.getIdentificador();
238         if(valor.equalsIgnoreCase(identificador)){
239             figura.add(f);
240         }
241     }
242     return figura;
243 }
```

Método “cambiarDatos()” “eliminarPieza()”

```
244 //Metodo cambiar datos, aqui solo se introducen dentro del ArrayList
245 //Tengo dos formas de hacerlo, uno comentado y otro no
246 public void cambiarDatos(String identificador, float alturaM, String material, int cantidad, int ano, String foto, String fabricante){
247     //String valor;
248     int n = 0, a=0;
249     /*
250     for(Figura f : generalList){
251         valor = f.getIdentificador();
252         if(valor.equalsIgnoreCase(identificador)){
253             //ha encontrado una figura la cual tiene el mismo identificador
254             f.setIdentificador(identificador);
255             f.setAlturaMetros(alturaM);
256             f.setMaterial(material);
257             f.setCantidad(cantidad);
258             f.setAño(ano);
259             f.setFoto(foto);
260             f.setFabricante(fabricante);
261         }
262         n+=1;
263     }
264     */
265     for(Figura tmp : generalList){
266         //Iteramos para ver en que indice se encuentra el identificador buscado
267         if(identificador.equalsIgnoreCase(tmp.getIdentificador())){
268             a=n;
269         }
270         n+=1;
271     }
272     Figura f = new Figura(identif: identificador, alturaMet: alturaM, material, cantidad, ano, foto, fabricante); // Crea una figura nueva
273     generalList.set(index: a, element: f); // Introduce los valores en el indice a para cambiarlos
274 }
275 // Metodo eliminar pieza pasandole el identificador
276 public void eliminarPieza(String pieza){
277     List<Figura> arrayRemove = new ArrayList<>();
278     for(Figura f : generalList){
279         if(pieza.equalsIgnoreCase(f.getIdentificador())){
280             //Significa que el identificador que se ha puesto es bueno y se elimina una pieza
281             arrayRemove.add(f); //Si se encuentra el identificador se mete en ArrayRemove
282         }
283     }
284     //Se elimina el valor introducido en arrayRemove
285     generalList.removeAll(arrayRemove);
286     //Se podria haber hecho mas eficiente pero queria probar el metodo removeAll
287 }
288 }
```

Método “listadoPorIdentificador()” “listadoPorAñoElIdentificador()” “listadoPorFabricante()”
“printfPantalla()” “imprimirConFormato()” “ponerPrimeraLetraEnMayuscula()”

```
////////////////////////////////// MENÚ LISTADOS ////////////////////////////////////

// Metodo en el que se ordena por identificador
public void listadoPorIdentificador(){
    Comparator<Figura> cId = Comparator.comparing(Figura::getIdentificador);
    Collections.sort(lista, generalList, cId);
}

//Metodo que se ordena por año y luego si tienen las figuras mismos años se ordenan por indetificador
public void listadoPorAñoElIdentificador(){
    Comparator<Figura> cAño = Comparator.comparing(Figura::getAño).reversed();
    Comparator<Figura> cId = Comparator.comparing(Figura::getIdentificador);
    Comparator<Figura> total = cAño.thenComparing(cId);
    Collections.sort(lista, generalList, total);
}

//Metodo que ordena por fabricante y si hay un mismo fabricante se ordena por año
public void listadoPorFabricanteYAño(){
    Comparator<Figura> cFabric = Comparator.comparing(Figura::getFabricante);
    Comparator<Figura> cAño = Comparator.comparing(Figura::getAño).reversed();
    Comparator<Figura> total = cFabric.thenComparing(cAño);
    Collections.sort(lista, generalList, total);
}

public String printfPantalla(List<Figura> f){
    for(Figura a : f){
        return String.format(" %s\t%f\t%s\t%d\t%d\t%s\t%s", args: a.getIdentificador(), args: a.getAlturaMetros(),
            args: a.getMaterial(), args: a.getCantidad(), args: a.getAño(), args: a.getFoto(), args: a.getFabricante());
    }
    return null;
}

//Metodo para imprimir por pantalla con printToScreen
public String[][] imprimirConFormato(){
    String[][] matriz = new String[generalList.size()][CAMPO];
    int n=0; //Numero que itera el primer indice ya que con el forEach no lo podemos saber
    for(Figura f : generalList){
        matriz[n] = f.valoresFigura();
        n++;
    }
    return matriz;
}

//Metodo para poner la primera letra en mayuscula, ya que lo necesito para el comparador
public String ponerPrimeraLetraMayuscula(String str){
    str = str.substring(beginIndex: 0, endIndex: 1).toUpperCase() + str.substring(beginIndex: 1);
    return str;
}
```


EJECUCION DEL PROGRAMA

Primera imagen que se ve al ejecutar el programa

```
Se ha leído el archivo binario correctamente
```

```
Menu Principal
```

```
=====
```

- 1.- Archivos
- 2.- Gestion de la Galeria
- 3.- Listados
- 4.- Salir

```
->
```

Entramos en la opción a.

```
Gestion Archivos
```

```
=====
```

- a.- Importar CSV
- b.- Exportar CSV
- c.- Exportar HTML
- d.- Volver al menu principal

```
->
```

Dentro de la pocion a

```
Método: importarCSV()
```

```
=====
```

```
La ruta donde estan los ficheros es C:\Users\ddecn\Desktop\datos_figuras\figuras.csv
```

```
Se ha leído correctamente el fichero
```

```
El numero de registros descartados es 0 de un total de 95 y su porcentaje de errores es de 0.00%
```

- a.- Importar CSV
- b.- Exportar CSV
- c.- Exportar HTML
- d.- Volver al menu principal

```
->
```

Ejecuto la opción b

```
El archivo se ha exportado correctamente
```

- a.- Importar CSV
- b.- Exportar CSV
- c.- Exportar HTML
- d.- Volver al menu principal

```
-> |
```

Ejecutamos c y también ponemos un error para saber qué hace

```
a.- Importar CSV
b.- Exportar CSV
c.- Exportar HTML
d.- Volver al menu principal
-> c#
Introduzca una opcion valida
a.- Importar CSV
b.- Exportar CSV
c.- Exportar HTML
d.- Volver al menu principal
-> c
Se ha exportado el fichero HTML correctamente

a.- Importar CSV
b.- Exportar CSV
c.- Exportar HTML
d.- Volver al menu principal
-> |
```

| | | | | | |
|---|---------|------------------|---------------------|-------|----------------------|
| This PC > Local Disk (C:) > Users > ddecn > Desktop > datos_figuras | | | | | Search datos_figuras |
| | Name | Date modified | Type | Size | |
| | figuras | 12/12/2022 16:14 | AceStream media ... | 6 KB | |
| | figuras | 12/12/2022 16:18 | CSV File | 5 KB | |
| | figuras | 12/12/2022 16:20 | Chrome HTML Do... | 11 KB | |

| GALERIA DE ARTE | | | | | | |
|-----------------|------------|-----------------|----------|------|-------------|------------|
| IDENTIFICADOR | ALTURA (M) | MATERIAL | CANTIDAD | AÑO | FOTO | FABRICANTE |
| 626 | 1.301000 | Goma | 2 | 2009 | 5261.png | Andrews |
| 645 | 0.103000 | Policarbonato | 4 | 2005 | 4695.png | Armstrong |
| 590 | 1.447000 | PVC | 1 | 1993 | 4491.png | Armstrong |
| 643 | 0.940000 | Aluminio | 2 | 1996 | 4505.png | Atkinson |
| 587 | 0.770000 | Aluminio | 3 | 2022 | 4483.png | Bell |
| 618 | 0.136000 | PVC | 4 | 1997 | 4957.png | Bell |
| 568 | 0.287000 | Pasta | 2 | 2022 | 5433.png | Bennett |
| 649 | 0.893000 | Vidrio | 4 | 2017 | 5489.png | Berry |
| 628 | 0.787000 | Níquel | 2 | 1997 | 4848.png | Brooks |
| 555 | 0.918000 | Ámbar | 2 | 2000 | 4519.png | Campbell |
| 565 | 1.298000 | Níquel | 1 | 1997 | 4910.png | Campbell |
| 578 | 0.973000 | Ámbar | 2 | 2018 | 5178.png | Carr |
| 588 | 0.239000 | Coral | 1 | 2006 | 5077.png | Carr |
| 570 | 1.435000 | Policarbonato | 1 | 1992 | 5051.png | Clarke |
| 646 | 0.312000 | Plástico | 2 | 2006 | 4717.png | Cole |
| 638 | 1.324000 | PVC | 3 | 2011 | 4567.png | Cook |
| 639 | 0.658000 | Madera y vidrio | 3 | 2005 | 4751.png | Cox |
| 651 | 1.087000 | Metal | 3 | 2004 | 5098.png | Cunningham |
| 573 | 0.992000 | Concha | 2 | 2002 | 5504.png | Davies |
| 586 | 0.238000 | Goma | 3 | 1989 | 4688.png | Davies |
| 567 | 0.764000 | Concha | 3 | 1982 | 4663.png | Davies |
| 630 | 0.661000 | Concha | 4 | 2014 | 4999.png | Davis |
| 625 | 0.835000 | Níquel | 1 | 2021 | 4460.png | Dean |
| 600 | 0.486000 | Cartón piedra | 3 | 2022 | 4813.png | Ellis |
| 2 | 1.200000 | diamante | 3 | 2019 | prueba2.png | Estudiante |
| 644 | 0.250000 | Aluminio | 2 | 1988 | 4635.png | Evans |
| 598 | 0.728000 | Vidrio | 4 | 2020 | 5259.png | Fisher |
| 621 | 1.277000 | Madera y vidrio | 4 | 1985 | 5114.png | Fisher |
| 596 | 0.893000 | Plástico | 3 | 2003 | 5195.png | Foster |
| 629 | 1.250000 | Cartón piedra | 3 | 1993 | 5325.png | Foster |

Vamos a ir probando cada opción

Salimos de la opción archivos y entramos en gestionar galería

```
1.- Archivos
2.- Gestion de la Galeria
3.- Listados
4.- Salir
-> 2

Menu Gestion de la Galeria
=====

a.- Añadir una figura al inventario
b.- Consultar datos de una figura
c.- Modificar datos de una figura
d.- Eliminar una figura del inventario
e.- Volver al menu principal

-> |
```

Entramos en la opción a con todos los errores que puede dar puestos y comprobados

```
Añadir una figura al inventario
=====

Introduzca un identificador para la pieza
->16
Ha introducido un identificador ya existente, introduzca otro por favor\n->

Introduzca un identificador para la pieza
->3

Introduce la altura en metros
-> (0.001000 <= numero <= 1.500000)2

Please write a number between 0.001000 and 1.500000

Introduce la altura en metros
-> (0.001000 <= numero <= 1.500000)-2

Please write a number between 0.001000 and 1.500000

Introduce la altura en metros
-> (0.001000 <= numero <= 1.500000)1.2

Introduce el material con el que esta construido en su mayoria
-> cobalto

Introduzca la cantidad de figuras de ese modelo
-> 3.5

Sorry, that is not a valid number. Please try again.

Introduzca la cantidad de figuras de ese modelo
-> 3

Introduzca el año de contruccion del modelo de la figura
-> (0 <= numero <= 2100)-3
```

```

Introduzca el año de construcción del modelo de la figura
-> (0 <= numero <= 2100)-3

Please write a number between 0 and 2100

Introduzca el año de construcción del modelo de la figura
-> (0 <= numero <= 2100)2366

Please write a number between 0 and 2100

Introduzca el año de construcción del modelo de la figura
-> (0 <= numero <= 2100)1258

Introduzca el nombre del archivo .PNG
-> valor
Introduzca el nombre con .png al final

Introduzca el nombre del archivo .PNG
-> valor.png

Escriba cual es el fabricante
-> yo
Se ha añadido la figura al inventario

a.- Añadir una figura al inventario
b.- Consultar datos de una figura
c.- Modificar datos de una figura
d.- Eliminar una figura del inventario
e.- Volver al menú principal

->

```

Entramos en la opción b y comprobamos que se ha añadido la figura con identificador 3.

```

Consulta de datos de una figura
=====
Introduzca el identificador de la figura
-> 3

La figura buscada es:
    Identificador 3
    Altura(M) 1.200000
    Material cobalto
    Cantidad      3
    Año      1258
    Foto valor.png
    Fabricante      Yo

a.- Añadir una figura al inventario
b.- Consultar datos de una figura
c.- Modificar datos de una figura
d.- Eliminar una figura del inventario
e.- Volver al menú principal

->

```

```

Introduzca el nuevo valor de la altura
-> (0.001000 <= numero <= 1.500000)1.3
Introduzca que valor quiere cambiar menos el identificador:
1- Altura
2- Material
3- Cantidad
4- Ano
5- Foto
6- Fabricante
s- Salir
-> 2
Introduzca el material nuevo
-> amatista
Introduzca que valor quiere cambiar menos el identificador:
1- Altura
2- Material
3- Cantidad
4- Ano
5- Foto
6- Fabricante
s- Salir
-> 4
Introduzca el nuevo ano de construccion de la figura
-> (0 <= numero <= 2100)2023
Introduzca que valor quiere cambiar menos el identificador:
1- Altura
2- Material
3- Cantidad
4- Ano
5- Foto
6- Fabricante
s- Salir
-> s
Quiere salir?[SsNn] s

a.- AÑadir una figura al invetario
b.- Consultar datos de una figura
c.- Modificar datos de una figuta
d.- Eliminar una figura del inventario
e.- Volver al menu principal

```

Como vemos se han cambiado los datos

```

Consulta de datos de una figura
=====
Introduzca el identificador de la figura
-> 3
La figura buscada es:
    Identificador 3
    Altura(M) 1.300000
    Material amatista
    Cantidad      3
    Ano      2023
    Foto valor.png
    Fabricante      Yo

a.- AÑadir una figura al invetario
b.- Consultar datos de una figura
c.- Modificar datos de una figuta
d.- Eliminar una figura del inventario
e.- Volver al menu principal

-> |

```

Eliminamos la figura 3 y la consultamos si sigue en el ArrayList

```
Eliminar Figura
=====
Introduzca el identificador de la pieza a eliminar
->3
La figura buscada es:%n
3          1.300000          amatista          3          2023          valor.png          Yo
Desea eliminar la pieza de verdad?
1- Si
2- No
-> (1 <= numero <= 2)1
Se ha eliminado la figura 3
a.- Añadir una figura al inventario
b.- Consultar datos de una figura
c.- Modificar datos de una figura
d.- Eliminar una figura del inventario
e.- Volver al menu principal

-> b

Consulta de datos de una figura
=====
Introduzca el identificador de la figura
-> 3
Ha introducido un identificador que no existe, se vuelve al menu de gestion

a.- Añadir una figura al inventario
b.- Consultar datos de una figura
c.- Modificar datos de una figura
d.- Eliminar una figura del inventario
e.- Volver al menu principal

-> |
```

Por último, probamos el listado

Probamos el listado por año e identificador

El listado de las figuras ordenadas por año e identificador es

| Identif | Altura (M) | Material | ID | Año | Foto | Fabricante |
|---------|------------|---------------|----|------|------------------|------------|
| 16 | 1.45 | piedra maciza | 3 | 2031 | momentoEpico.png | Yo |
| 568 | 0.287 | Pasta | 2 | 2022 | 5433.png | Bennett |
| 587 | 0.77 | Aluminio | 3 | 2022 | 4483.png | Bell |
| 597 | 0.213 | Plástico | 4 | 2022 | 4581.png | Ryan |
| 600 | 0.486 | Cartón piedra | 3 | 2022 | 4813.png | Ellie |
| 614 | 1.001 | Aluminio | 2 | 2022 | 5024.png | Thomson |
| 665 | 1.435 | Goma | 4 | 2022 | 5188.png | Moore |
| 611 | 0.65 | Madera | 3 | 2021 | 4776.png | Roberts |
| 625 | 0.835 | Niquel | 1 | 2021 | 4460.png | Dean |
| 652 | 0.992 | Madera | 1 | 2021 | 4564.png | Gardner |
| 558 | 1.382 | Coral | 2 | 2020 | 5063.png | Marshall |
| 598 | 0.728 | Vidrio | 4 | 2020 | 5259.png | Fisher |
| 624 | 1.393 | Ombar | 4 | 2020 | 5445.png | Harrison |
| 627 | 1.207 | Cartón piedra | 1 | 2020 | 5026.png | West |
| 666 | 1.173 | Plástico | 2 | 2020 | 5153.png | Mills |
| 2 | 1.2 | diamante | 3 | 2019 | prueba2.png | Estudiante |
| 605 | 0.941 | Madera | 3 | 2019 | 5129.png | Watts |
| 557 | 1.155 | Pasta | 3 | 2018 | 4735.png | Watts |
| 578 | 0.973 | Ombar | 2 | 2018 | 5178.png | Carr |
| 636 | 0.67 | Aluminio | 1 | 2017 | 4978.png | Williamson |
| 649 | 0.893 | Vidrio | 4 | 2017 | 5489.png | Berry |
| 559 | 1.178 | Concha | 3 | 2016 | 4734.png | Gordon |
| 576 | 0.367 | Pasta | 1 | 2016 | 4891.png | Webb |
| 642 | 0.854 | Madera | 3 | 2016 | 4822.png | Taylor |
| 648 | 0.199 | Concha | 1 | 2016 | 5213.png | Fox |
| 641 | 1.298 | Pasta | 3 | 2015 | 4758.png | George |
| 630 | 0.661 | Concha | 4 | 2014 | 4999.png | Davis |
| 658 | 0.332 | Madera | 1 | 2013 | 4993.png | Parker |
| 604 | 1.021 | Polycarbonato | 3 | 2012 | 5082.png | Rogers |
| 623 | 1.343 | Coral | 2 | 2012 | 5186.png | Lee |
| 638 | 1.324 | PVC | 3 | 2011 | 4567.png | Cook |
| 599 | 0.829 | Goma | 1 | 2010 | 5103.png | Moore |
| 566 | 1.464 | Aluminio | 1 | 2009 | 4593.png | Webb |
| 626 | 1.301 | Goma | 2 | 2009 | 5261.png | Andrews |
| 592 | 1.326 | Concha | 2 | 2008 | 4754.png | Moore |
| 632 | 1.463 | Plástico | 4 | 2008 | 5511.png | Pearson |
| 615 | 1.252 | Concha | 2 | 2007 | 4953.png | Webb |
| 588 | 0.239 | Coral | 1 | 2006 | 5077.png | Carr |

Segundo probamos el listado por identificador

El listado de las figuras ordenadas por identificador es

| Identif | Altura (M) | Material | ND | Ano | Foto | Fabricante | Yo |
|---------|------------|-----------------|----|------|------------------|---------------------------------|------------|
| 16 | 1.45 | piedra maciza | 3 | 2031 | momentoEpico.png | | Yo |
| 18 | 1.2 | cobalto | 2 | 1965 | a.png | Pruebahaceprimeraletramayuscula | |
| 2 | 1.2 | diamante | 3 | 2019 | prueba2.png | | Estudiante |
| 555 | 0.918 | Ombar | 2 | 2000 | 4519.png | | Campbell |
| 556 | 1.012 | Aluminio | 4 | 1981 | 4446.png | | Gordon |
| 557 | 1.155 | Pasta | 3 | 2018 | 4735.png | | Watts |
| 558 | 1.382 | Coral | 2 | 2020 | 5063.png | | Marshall |
| 559 | 1.178 | Concha | 3 | 2016 | 4734.png | | Gordon |
| 560 | 0.319 | FVC | 3 | 1987 | 5173.png | | Griffiths |
| 562 | 0.616 | Madera y vidrio | 1 | 1991 | 5135.png | | Wright |
| 563 | 1.464 | Vidrio | 3 | 1981 | 4475.png | | Gill |
| 564 | 0.264 | Ombar | 1 | 1989 | 5041.png | | Rose |
| 565 | 1.298 | NDquel | 1 | 1997 | 4910.png | | Campbell |
| 566 | 1.464 | Aluminio | 1 | 2009 | 4593.png | | Webb |
| 567 | 0.764 | Concha | 3 | 1982 | 4663.png | | Davies |
| 568 | 0.287 | Pasta | 2 | 2022 | 5433.png | | Bennett |
| 569 | 0.324 | PlOstico | 1 | 2005 | 5323.png | | Stone |
| 570 | 1.435 | Policarbonato | 1 | 1992 | 5051.png | | Clarke |
| 572 | 1.156 | PlOstico | 3 | 1981 | 4587.png | | Parker |
| 573 | 0.992 | Concha | 2 | 2002 | 5504.png | | Davies |
| 576 | 0.367 | Pasta | 1 | 2016 | 4891.png | | Webb |
| 577 | 0.721 | Vidrio | 2 | 1982 | 4828.png | | Shaw |
| 578 | 0.973 | Ombar | 2 | 2018 | 5178.png | | Carr |
| 579 | 0.683 | NDquel | 4 | 1988 | 4819.png | | Jackson |
| 581 | 0.813 | CartOn piedra | 1 | 1983 | 4518.png | | Marshall |
| 582 | 0.861 | Madera y vidrio | 3 | 1991 | 5398.png | | Gibson |
| 583 | 1.293 | Policarbonato | 2 | 1989 | 4672.png | | Wilson |
| 584 | 0.86 | Concha | 3 | 2002 | 5524.png | | Webb |
| 586 | 0.238 | Goma | 3 | 1989 | 4688.png | | Davies |
| 587 | 0.77 | Aluminio | 3 | 2022 | 4483.png | | Bell |
| 588 | 0.239 | Coral | 1 | 2006 | 5077.png | | Carr |
| 589 | 1.273 | Concha | 2 | 1986 | 5447.png | | Stevens |
| 590 | 1.447 | FVC | 1 | 1993 | 4491.png | | Armstrong |
| 592 | 1.326 | Concha | 2 | 2008 | 4754.png | | Moore |
| 593 | 1.046 | Ombar | 4 | 1983 | 4806.png | | Hunt |
| 594 | 0.643 | NDquel | 4 | 1999 | 4749.png | | Martin |
| 595 | 1.283 | CartOn piedra | 3 | 2003 | 5520.png | | Walsh |
| 596 | 0.893 | PlOstico | 3 | 2003 | 5195.png | | Foster |
| 597 | 0.213 | PlOstico | 4 | 2022 | 4581.png | | Ryan |

Y el ultimo listado es el de fabricante y año, he obligado a que la primera letra sea mayúscula siempre, así se hará ordenado.

El listado de las figuras ordenadas por identificador es

| Identif | Altura(M) | Material | ND | Ano | Foto | Fabricante | |
|---------|-----------|-----------------|----|------|------------------|---------------------------------|--|
| 16 | 1.45 | piedra maciza | 3 | 2031 | momentoEpico.png | Yo | |
| 18 | 1.2 | cobalto | 2 | 1965 | a.png | Pruebahaceprimeraletramayuscula | |
| 2 | 1.2 | diamante | 3 | 2019 | prueba2.png | Estudiante | |
| 555 | 0.918 | Ombar | 2 | 2000 | 4519.png | Campbell | |
| 556 | 1.012 | Aluminio | 4 | 1981 | 4446.png | Gordon | |
| 557 | 1.155 | Pasta | 3 | 2018 | 4735.png | Watts | |
| 558 | 1.382 | Coral | 2 | 2020 | 5063.png | Marshall | |
| 559 | 1.178 | Concha | 3 | 2016 | 4734.png | Gordon | |
| 560 | 0.319 | PVC | 3 | 1987 | 5173.png | Griffiths | |
| 562 | 0.616 | Madera y vidrio | 1 | 1991 | 5135.png | Wright | |
| 563 | 1.464 | Vidrio | 3 | 1981 | 4475.png | Gill | |
| 564 | 0.264 | Ombar | 1 | 1989 | 5041.png | Rose | |
| 565 | 1.298 | NDquel | 1 | 1997 | 4910.png | Campbell | |
| 566 | 1.464 | Aluminio | 1 | 2009 | 4593.png | Webb | |
| 567 | 0.764 | Concha | 3 | 1982 | 4663.png | Davies | |
| 568 | 0.287 | Pasta | 2 | 2022 | 5433.png | Bennett | |
| 569 | 0.324 | PlOstico | 1 | 2005 | 5323.png | Stone | |
| 570 | 1.435 | Polycarbonato | 1 | 1992 | 5051.png | Clarke | |
| 572 | 1.156 | PlOstico | 3 | 1981 | 4587.png | Parker | |
| 573 | 0.992 | Concha | 2 | 2002 | 5504.png | Davies | |
| 576 | 0.367 | Pasta | 1 | 2016 | 4891.png | Webb | |
| 577 | 0.721 | Vidrio | 2 | 1982 | 4828.png | Shaw | |
| 578 | 0.973 | Ombar | 2 | 2018 | 5178.png | Carr | |
| 579 | 0.683 | NDquel | 4 | 1988 | 4819.png | Jackson | |
| 581 | 0.813 | CartOn piedra | 1 | 1983 | 4518.png | Marshall | |
| 582 | 0.861 | Madera y vidrio | 3 | 1991 | 5398.png | Gibson | |
| 583 | 1.293 | Polycarbonato | 2 | 1989 | 4672.png | Wilson | |
| 584 | 0.86 | Concha | 3 | 2002 | 5524.png | Webb | |
| 586 | 0.238 | Goma | 3 | 1989 | 4688.png | Davies | |
| 587 | 0.77 | Aluminio | 3 | 2022 | 4483.png | Bell | |
| 588 | 0.239 | Coral | 1 | 2006 | 5077.png | Carr | |
| 589 | 1.273 | Concha | 2 | 1986 | 5447.png | Stevens | |
| 590 | 1.447 | PVC | 1 | 1993 | 4491.png | Armstrong | |
| 592 | 1.326 | Concha | 2 | 2008 | 4754.png | Moore | |
| 593 | 1.046 | Ombar | 4 | 1983 | 4806.png | Hunt | |
| 594 | 0.643 | NDquel | 4 | 1999 | 4749.png | Martin | |
| 595 | 1.283 | CartOn piedra | 3 | 2003 | 5520.png | Walsh | |
| 596 | 0.893 | PlOstico | 3 | 2003 | 5195.png | Poster | |
| 597 | 0.213 | PlOstico | 4 | 2022 | 4581.png | Ryan | |

Y por último salimos del programa dándole a opción 4 y a la s.

```

| 662 | 0.14 | Polycarbonato | 4 | 1982 | 5306.png | Robinson |
| 664 | 1.423 | Aluminio | 3 | 1998 | 4724.png | White |
| 665 | 1.435 | Goma | 4 | 2022 | 5188.png | Moore |
| 666 | 1.173 | PlOstico | 2 | 2020 | 5153.png | Mills |
| 988 | 1.4 | cobre | 3 | 2000 | figuras.png | yo |

a.- Listado por identificador
b.- Listado por año e identificador
c.- Listado por fabricante y año
d.- Volver al menu principal
-> d
Ha salido de la opcion listados
No ha introducido ninguna de las posibilidades

1.- Archivos
2.- Gestion de la Galeria
3.- Listados
4.- Salir
-> 4

¿Quiere salir del programa? s/n[SsNn] s
Ha salido del programa

+-----+
| User           : ddecn          |
+-----+
| Task finished at :    16.41.27 |
+-----+
BUILD SUCCESSFUL (total time: 26 minutes 56 seconds)

```

