



# **PROGRAMACIÓN ORIENTADA A OBJETOS**

## **TEMA 6 FLUJOS Y ARCHIVOS**

### **ACTIVIDAD 6.2**

**GRUPO:** 2g2A

### **NOMBRE DE LOS ALUMNOS:**

- ENRIQUEZ MONTALVO RIGOBERTO
- GIL RODRIGUEZ JONATHAN
- GUTIERREZ CRUZ ÁNGEL DE JESÚS
- MORALES VAZQUEZ JUAN DIEGO
- SOSA FIGUEROA BENJAMÍN DE JESÚS

**PROFESORA:** PATRICIA QUITL GONZALEZ

**FECHA DE ENTREGA:** 29/05/2023

**HORA DE CLASE:** 11:00-12:00



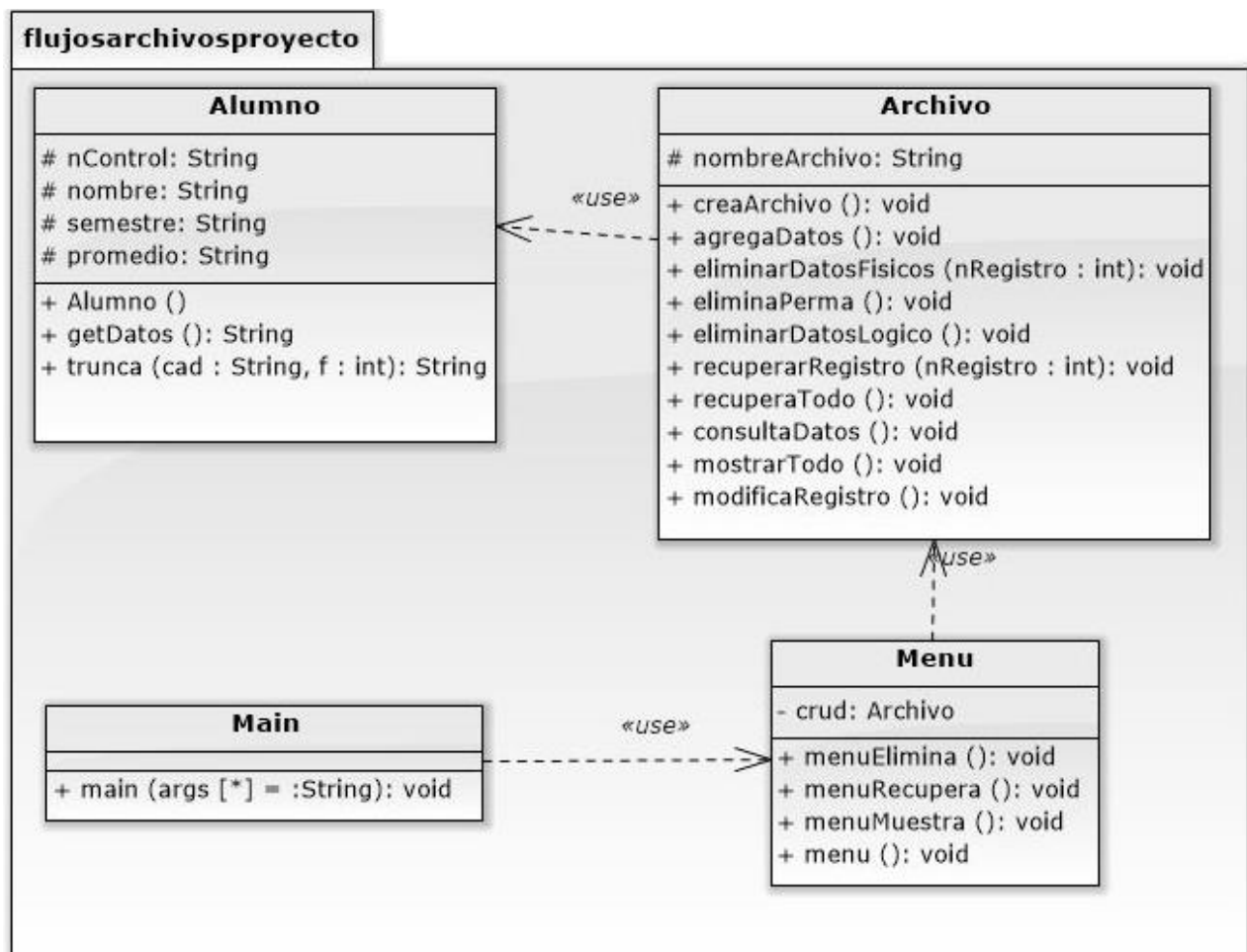
## PROYECTO FINAL

### Definición del problema

Elaborar un proyecto en Java que permita las operaciones básicas sobre un archivo de acceso aleatorio, con la estructura de los registros que les sea conveniente a los estudiantes, facilitar las operaciones mediante un menú de opciones:

1. Crear archivo.
2. Agregar datos.
3. Mostrar el contenido del archivo.
4. Consultar un registro determinado.
5. Eliminar de manera lógica un registro.
6. Recuperar todos los registros eliminados de manera lógica.
7. Eliminar definitivamente los registros eliminados lógicamente.

### Diseño de la solución mediante diagrama de clases (UML)



## Códigos nítidos

### Clase Alumno:

```

1  package flujosarchivosproyecto;
2
3  import jopi.JOPI;
4
5  public class Alumno {
6
7      protected String nControl, nombre, semestre, promedio;
8
9      public Alumno() {
10
11          this.nControl = trunca(cad:JOPI.cadena(msg:"No Control? ") , f: 10); // 8
12          this.nombre = trunca(cad:JOPI.cadena(msg:"Nombre? ") , f: 20); // 20
13          this.semestre = trunca(cad:JOPI.cadena(msg:"Semestre? ") , f: 3); // 2
14          this.promedio = trunca(cad:JOPI.cadena(msg:"Promedio? ") , f: 3); // 3
15      }
16
17      public String getDatos() {
18          String msg = this.nControl + "\n";
19          msg += this.nombre + "\n";
20          msg += this.semestre + "\n";
21          msg += this.promedio+"\n";
22
23          return msg;
24      }
25
26      public String trunca(String cad , int f) {
27          String msg = "";
28
29          msg = cad+ " ";
30          return msg.substring(beginIndex: 0, endIndex: f);
31      }
32
33  }

```

## Clase Archivo:

```

1 package flujosarchivosproyecto;
2
3 import java.io.File;
4 import java.io.FileNotFoundException;
5 import java.io.IOException;
6 import java.io.RandomAccessFile;
7 import jopi.JOPI;
8 import salida.JOPIS;
9
10 public class Archivo {
11
12     protected String nombreArchivo;
13
14     public void creaArchivo() {
15         nombreArchivo = JOPI.cadena(msg:"Ingresa el nombre del archivo");
16         File f = new File(nombreArchivo + ".dat");
17         f.delete();
18     }
19
20     public void agregaDatos() {
21         RandomAccessFile archivo = null;
22
23         try {
24             File f = new File(nombreArchivo + ".dat");
25             archivo = new RandomAccessFile(f, mode:"rw");
26             archivo.seek(pos:archivo.length());
27
28             Archivo arch = new Archivo();
29
30             String alum = new Alumno().getDatos();
31             archivo.writeBytes("\0"+"n"+alum);
32             archivo.writeBytes("\n");
33             archivo.close();
34         } catch (FileNotFoundException e) {
35             System.err.println("Error: el archivo no se pudo crear");
36         } catch (IOException e) {
37             System.err.println("Error: no se pudo cerrar el archivo");
38         }
39     }
40 }
41

```

```

42 public void eliminarDatosFisico(int nRegistro) {
43
44     try {
45         File f = new File(nombreArchivo + ".dat");
46         File f2 = new File("copia_"+nombreArchivo+".dat");
47         f2.delete();
48         RandomAccessFile archivo2 = new RandomAccessFile(f2, mode: "rw");
49         archivo2.seek(pos:0);
50         int i = 1; //contador de espacios
51         String guardado = "";
52         RandomAccessFile archivo = new RandomAccessFile(f, mode: "rw");
53         archivo.seek(pos:0);
54
55         while ((guardado = archivo.readLine()) != null) {
56
57             if (i != nRegistro) {
58                 archivo2.writeBytes(guardado + "\n");
59             }
60
61             if (guardado.isBlank()) {
62                 i++;
63             }
64         }
65         // Cerrar los archivos
66         archivo.close();
67         archivo2.close();
68
69         Thread.sleep(millis: 1000);
70
71         if (f.delete()) {
72             f2.renameTo(dest: f);
73         } else {
74             System.err.println(x: "Error: aaaaa no se pudo eliminar el archivo original");
75         }
76     } catch (FileNotFoundException e) {
77         System.err.println(x: "Error el archivo no se pudo crear");
78
79     } catch (IOException e) {
80         System.err.println(x: "Error no se pudo cerrar el archivo");
81     } catch (InterruptedException ex) {
82         System.err.println(x: "Se murio");
83     }
84
85     return;
86 }
87
88 public void eliminaPerma(){
89
90     try {
91         File f = new File(nombreArchivo + ".dat");
92         int i = 1; //contador de espacios
93         int p = 1;
94         String guardado = "";
95         RandomAccessFile archivo = new RandomAccessFile(f, mode: "rw");
96         archivo.seek(pos:0);

```

```

96     int arr[] = new int[100001];
97     int cont=0;
98     while ((guardado = archivo.readLine()) != null) {
99
100         if(guardado.equals(anObject: "1"))
101             arr[cont++] = i;
102
103         if (guardado.isBlank())
104             i++;
105     }
106
107     archivo.close();
108     for(int c=cont-1; c>=0; c--){
109         Thread.sleep(millis: 1000);
110         System.out.println(arr[c]);
111         this.eliminarDatosFisico(arr[c]);
112     }
113     // Cerrar los archivos
114
115     } catch (FileNotFoundException e) {
116         System.err.println(x: "Error el archivo no se pudo crear");
117     } catch (IOException e) {
118         System.err.println(x: "Error no se pudo cerrar el archivo");
119     } catch (InterruptedException ex) {
120         System.out.println(x: "Se murio");
121     }
122 }
123
124 public void eliminarDatosLogico() {
125     //SOLICITAR EL NUMERO DE REGISTRO A ELIMINAR
126     int nRegistro = JOPI.entero(msg: "Ingresa el numero de registro que desea eliminar"); //CANTIDAD DE SALTOS DE LINEA
127
128     //ABRIR EL ARCHIVO ORIGINAL Y COLOCARSE AL INICIO
129     try {
130         File f = new File(nombreArchivo + ".dat");
131         int i = 1; //contador de espacios
132
133         int p = 1;
134         String guardado = "";
135         RandomAccessFile archivo = new RandomAccessFile(f, mode: "rw");
136         archivo.seek(pos: 0);
137
138         if (nRegistro == 1) {
139             archivo.writeBytes(s: "1");
140             i++;
141         }
142         while ((guardado = archivo.readLine()) != null) {
143
144             if (guardado.isBlank()) {
145                 i++;
146             }
147
148             while ((i == nRegistro && i != 1) && p == 1) {
149                 archivo.writeBytes(s: "1");
150                 archivo.readLine();
151                 guardado = archivo.readLine();
152                 p++;
153             }
154
155             // Cerrar los archivos
156             archivo.close();
157         } catch (FileNotFoundException e) {
158             System.err.println(x: "Error el archivo no se pudo crear");
159         } catch (IOException e) {
160             System.err.println(x: "Error no se pudo cerrar el archivo");
161         }
162     }
163     return;
164 }

```

```

165
166 public void recuperarRegistro(int nRegistro) {
167
168     //ABRIR EL ARCHIVO ORIGINAL Y COLOCARSE AL INICIO
169     try {
170         File f = new File(nombreArchivo + ".dat");
171         int i = 1; //contador de espacios
172         int p = 1;
173         String guardado = "";
174         RandomAccessFile archivo = new RandomAccessFile(file: f, mode: "rw");
175         archivo.seek(pos:0);
176
177         if (nRegistro == 1) {
178             archivo.writeBytes(s: "0");
179             i++;
180         }
181         while ((guardado = archivo.readLine()) != null) {
182
183             if (guardado.isBlank()) {
184                 i++;
185             }
186
187             while ((i == nRegistro && i != 1) && p == 1) {
188                 archivo.writeBytes(s: "0");
189                 archivo.readLine();
190                 guardado = archivo.readLine();
191                 p++;
192             }
193
194         }
195         // Cerrar los archivos
196         archivo.close();
197
198     } catch (FileNotFoundException e) {
199         System.err.println(x: "Error el archivo no se pudo crear");
200     } catch (IOException e) {
201         System.err.println(x: "Error no se pudo cerrar el archivo");
202         System.err.println(x: "Error no se pudo cerrar el archivo");
203     }
204 }
205
206
207 public void recuperaTodo(){
208
209     try {
210         File f = new File(nombreArchivo + ".dat");
211         int i = 1; //contador de espacios
212         String guardado;
213         RandomAccessFile archivo = new RandomAccessFile(file: f, mode: "rw");
214         archivo.seek(pos:0);
215         int arr[] = new int[100001];
216         int cont=0;
217         while ((guardado = archivo.readLine()) != null) {
218
219             if(guardado.equals(anObject: "1"))
220                 arr[cont++] = i;
221
222             if (guardado.isBlank())

```

```

223         i++;
224     }
225
226     archivo.close();
227     for(int c=0; c < cont; c++){
228         Thread.sleep(milliseconds: 1000);
229         System.out.println(arr[c]);
230         this.recuperarRegistro(arr[c]);
231     }
232     // Cerrar los archivos
233
234     } catch (FileNotFoundException e) {
235         System.err.println("Error el archivo no se pudo crear");
236     } catch (IOException e) {
237         System.err.println("Error no se pudo cerrar el archivo");
238     } catch (InterruptedException ex) {
239         System.out.println("Se murio");
240     }
241 }
242
243 public void consultaDatos() {
244
245     int nRegistro = JOPI.entero(msg:"Ingresa el numero de registro que desea consultar"); //CANTIDAD DE SALTO DE LINEA
246
247     try {
248         File f = new File(nombreArchivo + ".dat");
249         RandomAccessFile archivo = new RandomAccessFile(file: f, mode: "r");
250         archivo.seek(pos:0);
251         nRegistro--;
252         System.out.println(" : archivo.length() ");
253
254         if( (nRegistro*43) >= archivo.length() ){
255             JOPI.mensaje(msg:"El registro no existe!!");
256             return;
257         }
258
259         archivo.seek((nRegistro * 43));
260
261         String linea = archivo.readLine();
262         System.out.println("linea = " + linea);
263         if(linea.equals(anObject: "0")){
264
265             linea = archivo.readLine()+"\n";
266             linea += archivo.readLine()+"\n";
267             linea += archivo.readLine()+"\n";
268             linea += archivo.readLine();
269
270             JOPI.mensaje(msg:linea);
271         }
272         else
273             JOPI.mensaje(msg:"El registro no existe");
274
275         // Cerrar los archivos
276         archivo.close();
277     } catch (FileNotFoundException e) {
278         System.err.println("Error el archivo no se pudo crear");
279     } catch (IOException e) {
280         System.err.println("Error no se pudo cerrar el archivo");
281     }
282 }
283
284 public void mostrarTodo(){
285
286     try {
287         File f = new File(nombreArchivo + ".dat");
288         RandomAccessFile archivo = new RandomAccessFile(file: f, mode: "r");
289         archivo.seek(pos:0);
290         System.out.println(" : archivo.length() ");
291         String linea , todo="";
292
293         if(archivo.length() == 0)

```



```

293         System.out.println(x: "El archivo esta vacio");
294     else
295     {
296         while( (linea = archivo.readLine()) != null)
297             todo += linea+"\n";
298
299         JOPIS.mensaje(msg:todo);
300     }
301     // Cerrar los archivos
302     archivo.close();
303 } catch (FileNotFoundException e) {
304     System.err.println(x: "Error el archivo no se pudo crear");
305 } catch (IOException e) {
306     System.err.println(x: "Error no se pudo cerrar el archivo");
307 }
308
309
310 }
311
312 public void modificaRegistro() {
313
314     try {
315         int nRegistro = JOPI.entero(msg:"Registro a modificar?");
316         File f = new File(nombreArchivo + ".dat");
317         File f2 = new File("copia_"+nombreArchivo+".dat");
318         f2.delete();
319         RandomAccessFile archivo2 = new RandomAccessFile(f2, mode: "rw");
320         archivo2.seek(pos:0);
321         int i = 1; //contador de espacios
322         String guardado = "";
323         RandomAccessFile archivo = new RandomAccessFile(f, mode: "rw");
324         archivo.seek(pos:0);
325
326         while ((guardado = archivo.readLine()) != null) {
327
328             if (i != nRegistro)
329                 archivo2.writeBytes(guardado + "\n");
330
331             else{
332                 archivo2.writeBytes("0\n"+new Alumno().getDatos());
333                 guardado = archivo.readLine();
334                 guardado = archivo.readLine();
335                 guardado = archivo.readLine();
336                 guardado = archivo.readLine();
337                 i++;
338             }
339
340             if (guardado.isBlank())
341                 i++;
342
343         }
344         // Cerrar los archivos
345         archivo.close();
346         archivo2.close();
347
348         Thread.sleep(millis: 1000);
349
350         if (f.delete())
351             f2.renameTo(dest: f);
352         else
353             System.err.println(x: "Error: aaaa no se pudo eliminar el archivo original");

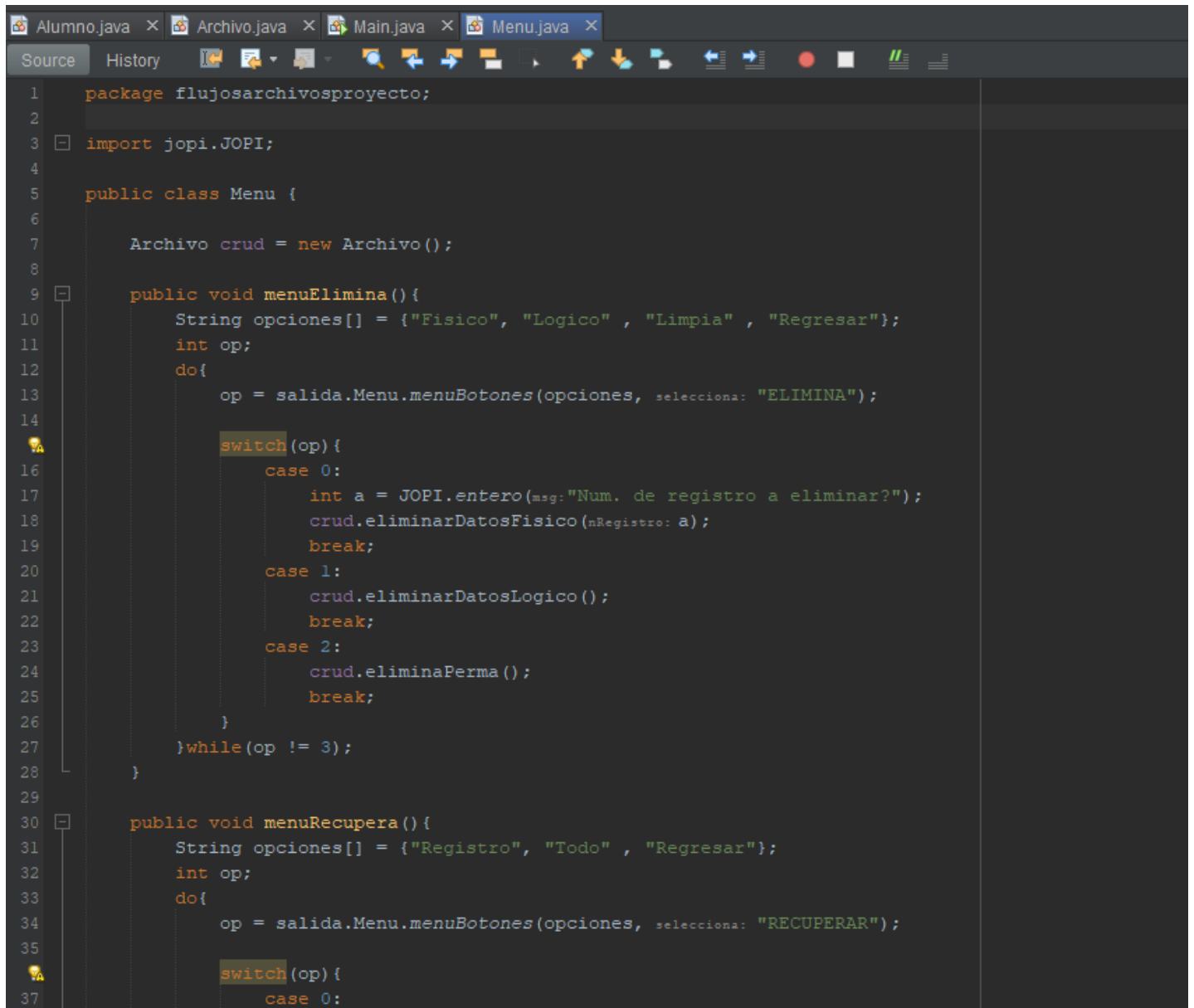
```

```

354     } catch (FileNotFoundException e) {
355         System.err.println(x: "Error el archivo no se pudo crear");
356     } catch (IOException e) {
357         System.err.println(x: "Error no se pudo cerrar el archivo");
358     } catch (InterruptedException ex) {
359         System.err.println(x: "Se murio");
360     }
361
362 }
363
364 }

```

## Clase Menu:



```

Alumno.java x Archivo.java x Main.java x Menu.java x
Source History
1 package flujosarchivosproyecto;
2
3 import jopi.JOPI;
4
5 public class Menu {
6
7     Archivo crud = new Archivo();
8
9     public void menuElimina() {
10         String opciones[] = {"Fisico", "Logico", "Limpia", "Regresar"};
11         int op;
12         do {
13             op = salida.Menu.menuBotones(opciones, selecciona: "ELIMINA");
14
15             switch (op) {
16                 case 0:
17                     int a = JOPI.entero(msg: "Num. de registro a eliminar?");
18                     crud.eliminarDatosFisico(nRegistro: a);
19                     break;
20                 case 1:
21                     crud.eliminarDatosLogico();
22                     break;
23                 case 2:
24                     crud.eliminaPerma();
25                     break;
26             }
27         } while (op != 3);
28     }
29
30     public void menuRecupera() {
31         String opciones[] = {"Registro", "Todo", "Regresar"};
32         int op;
33         do {
34             op = salida.Menu.menuBotones(opciones, selecciona: "RECUPERAR");
35
36             switch (op) {
37                 case 0:

```

```

38         int b = JOPI.entero(msg:"Ingresa el numero de registro que desea recuperar");
39         crud.recuperarRegistro(nRegistro: b);
40         break;
41     case 1:
42         crud.recuperaTodo();
43         break;
44     }
45     }while(op != 2);
46 }
47
48 public void menuMuestra(){
49     String opciones[] = {"Registro", "Todo" , "Regresar"};
50     int op;
51     do{
52         op = salida.Menu.menuBotones(opciones, selecciona: "MOSTRAR");
53
54         switch (op) {
55             case 0:
56                 crud.consultaDatos();
57                 break;
58             case 1:
59                 crud.mostrarTodo();
60                 break;
61         }
62     }while(op != 2);
63 }
64
65 public void menu() {
66
67     crud.creaArchivo();
68
69     String opciones[] = {"Agregar Datos" , "Consultar" , "Modifica" , "Elimina", "Recupera" ,"Terminar"};
70     int opcion;
71     do {
72         opcion = salida.Menu.menuBotones(opciones, selecciona: "Menu principal");
73
74         switch (opcion) {
75             case 0:
76                 crud.agregaDatos();
77                 break;
78             case 1:
79                 this.menuMuestra();
80                 break;
81             case 2:
82                 crud.modificaRegistro();
83                 break;
84             case 3:
85                 this.menuElimina();
86                 break;
87             case 4:
88                 this.menuRecupera();
89                 break;
90         }
91
92     } while (opcion != 5);
93 }
94
95 }
96


```

## Clase Main:

```
Alumno.java x Archivo.java x Main.java x Menu.java x
Source History
1 package flujosarchivosproyecto;
2
3 public class Main {
4     public static void main(String[] args) {
5         Menu menu = new Menu();
6         menu.menu();
7     }
8 }
9
```

## Casos de prueba

Input

 Ingresa el nombre del archivo

Tester


OK Cancel

OPCIONES

Menu principal

Agregar Datos Consultar Modifica Elimina Recupera Terminar


Input

 No Control?

2201102

OK Cancel


Input

 Nombre?


Benjamin Sosa

OK Cancel

Input

 **Semestre?**

Input


 **Promedio?**

Tester.dat

Archivo Editar Ver

0  
2201102  
Benjamin Sosa  
2  
80


Input

 **Nombre?**


OPCIONES

**MOSTRAR**


Input

 **Ingresa el numero de registro que desea consultar**

Message


 2201102  
Benjamin Sosa  
2  
80

Message


 0  
2201102  
Benjamin Sosa  
2  
80

0  
22505  
Diego Morales  
2  
100

Input


 **Registro a modificar?**

Input

**No Control?**  
  


OK Cancel

Input

**Nombre?**  
  


OK Cancel

Input

**Semestre?**  
  

OK Cancel

Input

**Promedio?**  
  

OK Cancel

Tester.dat

Archivo Editar Ver

0  
22505  
Angel Gutierrez  
2  
58


0  
22505  
Diego Morales  
2  
100

OPCIONES

**ELIMINA**  

Fisico Logico Limpia Regresar

Input

**Num. de registro a eliminar?**  
  


OK Cancel

Tester.dat

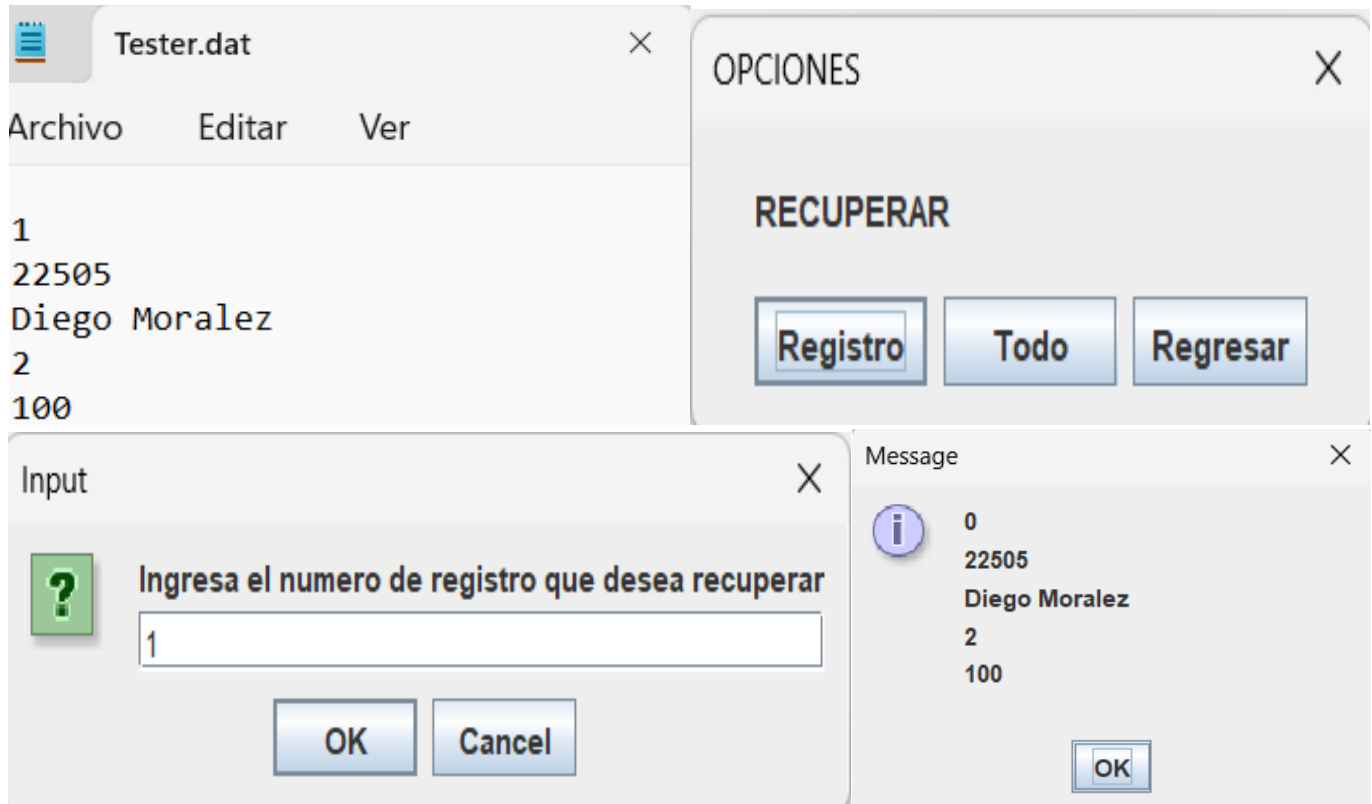
Archivo Editar Ver

0  
22505  
Diego Morales  
2  
100

Input

**Ingresa el numero de registro que desea eliminar**  
  

OK Cancel



## Conclusiones

Enríquez Montalvo Rigoberto: La realización de este proyecto me permitió adquirir nuevos conocimientos con base al manejo de archivos utilizando las clases "File" y "RandomAccessFile" que Java nos proporciona para poder acceder, manipular y modificar, según lo requiera el caso, el archivo presente. Por su parte, las excepciones fueron de la mano con este proyecto, pues conforme se iba trabajando se dieron a relucir ciertos errores, los cuales se pudieron visualizar gracias al uso de las excepciones. Del mismo modo, se logró implementar un CRUD más completo al agregar otros métodos, como lo son: la eliminación lógica, la modificación de archivos, etc. Si bien, ha sido un proyecto donde se pudieron poner en práctica varios conocimientos adquiridos a lo largo de esta materia.

Gil Rodriguez Jonathan: Durante la elaboración del proyecto, pude darme cuenta que todos teníamos diferentes soluciones para un mismo problema, sin embargo, aprendí que algunas soluciones pueden ser más eficientes que otras, dependiendo de cada programador. Al utilizar la clase File y RandomAccessFile y sus distintos métodos me di cuenta que es sumamente importante manejarlos con cuidado, ya que a veces se producen excepciones de las cuáles muchas veces no estamos conscientes del porqué, y esto es debido a que al manejar archivos también entra el sistema operativo, por lo que ya no es lo mismo que trabajar con la memoria principal.

También, al manejar excepciones pude reforzar los conocimientos sobre cómo ordenar las excepciones y para que sirve cada una, fue de gran ayuda ya que me permitió visualizar cuáles eran los problemas que tenía. Puedo concluir que aprendí mucho como manejar archivos, al igual que distintos métodos para hacer posible un CRUD completo, además, gracias a los conocimientos adquiridos anteriormente sobre la programación orientada a objetos logramos hacer un código mucho más eficiente y ordenado.

Gutiérrez Cruz Ángel de Jesús: Gracias a este proyecto pude darme cuenta de la complejidad del manejo de la información dentro de un documento de texto y que al manipularlos existe una amplia cantidad de errores y excepciones que pueden ocurrir durante la ejecución del programa. Igual pude observar el comportamiento de los bloques y que es más sencillo dejar las escrituras de forma lineal, ya que al estar de esta forma la información, es más sencillo manipularlos, debido a que nos evitamos el hacer unas cuantas operaciones matemáticas y el limitar el número de caracteres que pueden existir dentro de cada bloque que conforman el archivo.

Morales Vázquez Juan Diego: En este proyecto desarrolle diferentes maneras de trabajar con los archivos usando lenguaje Java.

A lo largo de esta práctica aprendí diferentes algoritmos para lograr desarrollar las operaciones básicas de la información aplicadas a su manejo con archivos.

Vi diferentes formas de desarrollar esta práctica, algunas más efectivas o más sencillas que otras, por lo que aprendí más sobre los recursos que Java nos ofrece para manejar archivos, como RandomAccessFile y File.

También aplique muchos conocimientos pasados como las excepciones ya que, al trabajar con archivos y rutas de memoria, pueden producirse muchas excepciones a tratar.



Al hacer los diferentes algoritmos que use en el desarrollo del CRUD, practique mucho el manejo de estos por lo que me siento seguro de poder seguir trabajando con estos en un futuro pues es algo vital en el desarrollo, donde como programador debo saber manejar la información dentro de una computadora.

Sosa Figueroa Benjamín de Jesús: Gracias a este proyecto, he podido profundizar en el manejo de archivos, comprendiendo lo que realmente hace un sistema operativo o más específicamente el explorador de archivos. Logré realizar una papelera lógica que permite visualizar un archivo eliminado y con la facilidad de poder recuperarlo, al igual que nuestros equipos lo hacen.

Me permitió adentrarme más en el manejo de archivos. A diferencia del eliminado físico, que era un poco más sencillo, en este caso nos enfrentamos a algunas especificaciones con nuestros procesadores y el sistema operativo. Buscamos alternativas como el "Thread.Sleep", que personalmente me ayudó mucho, ya que el código desarrollado en mi máquina no funcionaba correctamente. Aprendí a utilizar nuevas clases y, a diferencia del eliminado físico, ahora tenemos un método "mostrar" que permite una mayor comodidad al visualizar la información contenida. Incluso, a pesar de no haberse logrado, existe una forma de "encriptar" la información contenida en un archivo, siendo esta una nueva herramienta que brinda más sentido a la opción de mostrar.

Lo que más me dificultó, sin lugar a dudas, fue la modificación. Al intentar modificar el archivo, muchas veces mi lógica no sabía cómo realizarlo. Gracias al gran equipo con el que estoy trabajando, logré tener otra perspectiva para solucionar los inconvenientes presentados y así encontrar una mejor alternativa para el proyecto.