

Universidad Nacional de San Agustín
Escuela Profesional de Ingeniería de Sistemas
Fundamentos de Programación II
Tema N° 21:
Archivos II

Nombre: Jhonatan Benjamin Mamani Céspedes

CUI: 20232188

Link de GitHub: <https://github.com/JBenjamin01/fp2-24b>

Ejercicio 1:

Leer e imprimir en pantalla los datos (descripción, cantidad y números) escritos en el archivo del Ejemplo (archivo con estructura). Primero leer los números a un ArrayList, luego imprimirlos

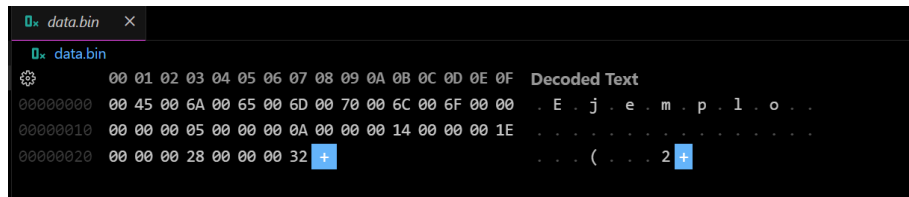
Clase FileWriter.java:

```
1  import java.io.*;
2
3  public class FileWriter {
4      public static void main(String[] args) {
5          String fileName = "data.bin";
6          String tableName = "Ejemplo";
7          int numRecords = 5;
8          int[] numbers = {10, 20, 30, 40, 50};
9
10         try (DataOutputStream fileOut = new DataOutputStream(new FileOutputStream(fileName))) {
11             for (char ch : tableName.toCharArray())
12                 fileOut.writeChar(ch);
13
14             fileOut.writeChar('\0');
15
16             fileOut.writeInt(numRecords);
17
18             for (int number : numbers)
19                 fileOut.writeInt(number);
20
21             System.out.println("Archivo generado correctamente: " + fileName);
22         } catch (IOException e) {
23             System.err.println("Error al escribir el archivo: " + e.getMessage());
24         }
25     }
26 }
```

Consola

```
Archivo generado correctamente: data.bin
```

Vista hexadecimal del archivo binario data.bin:



Clase Reader.java:

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Reader {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         System.out.print("Ingrese el nombre del archivo a leer: ");
8         String fileName = sc.nextLine();
9
10        try (DataInputStream fileIn = new DataInputStream(new FileInputStream(fileName))) {
11            StringBuilder nombreTablaBuilder = new StringBuilder();
12            char ch;
13            while ((ch = fileIn.readChar()) != 0) {
14                nombreTablaBuilder.append(ch);
15            }
16            String nombreTabla = nombreTablaBuilder.toString();
17
18            int numRecords = fileIn.readInt();
19
20            ArrayList<Integer> numeros = new ArrayList<>();
21            for (int i = 0; i < numRecords; i++)
22                numeros.add(fileIn.readInt());
23
24            System.out.println("Nombre de la tabla: " + nombreTabla);
25            System.out.println("Número de registros: " + numRecords);
26            System.out.println("Números leídos:");
27            for (int n : numeros)
28                System.out.println(n);
29
30        } catch (IOException e) {
31            System.err.println("Error al leer el archivo: " + e.getMessage());
32        }
33    }
34 }
```

Consola:

```
Ingrese el nombre del archivo a leer: data.bin
Nombre de la tabla: Ejemplo
Número de registros: 5
Números leídos:
10
20
30
40
50
```

Ejercicio 2:

Generar un conjunto de N enteros aleatorio en intervalo [x..y] y almacenarlos en un archivo binario, luego leerlos, ordenarlos y almacenarlos en otro archivo binario

Clase Aleatorios.java:

```
1  import java.io.*;
2  import java.util.*;
3
4  public class Aleatorios {
5      public static void main(String[] args) {
6          Scanner sc = new Scanner(System.in);
7
8          try {
9              System.out.print("Ingrese el nombre del archivo para guardar los números aleatorios: ");
10             String archivoAleatorios = sc.nextLine();
11
12             System.out.print("Ingrese el nombre del archivo para guardar los números ordenados: ");
13             String archivoOrdenados = sc.nextLine();
14
15             System.out.print("Ingrese el número de enteros (N): ");
16             int n = sc.nextInt();
17
18             System.out.print("Ingrese el valor mínimo del intervalo (x): ");
19             int x = sc.nextInt();
20
21             System.out.print("Ingrese el valor máximo del intervalo (y): ");
22             int y = sc.nextInt();
23
24             if (x > y) {
25                 System.out.println("El valor mínimo (x) no puede ser mayor que el máximo (y).");
26                 return;
27             }
28
29             Random r = new Random();
30             List<Integer> numeros = new ArrayList<>();
31             for (int i = 0; i < n; i++) {
32                 numeros.add(r.nextInt(y - x + 1) + x);
33             }
34
35             try (DataOutputStream aleatorios = new DataOutputStream(new FileOutputStream(archivoAleatorios))) {
36                 for (int num : numeros) {
37                     aleatorios.writeInt(num);
38                 }
39             } catch (Exception e) {
40                 System.err.println(e.getMessage());
41             }
42
43             System.out.println("Números aleatorios guardados en el archivo: " + archivoAleatorios);
44
45             List<Integer> numerosLeidos = new ArrayList<>();
46             try (DataInputStream randomFile = new DataInputStream(new FileInputStream(archivoAleatorios))) {
47                 while (randomFile.available() > 0) {
48                     numerosLeidos.add(randomFile.readInt());
49                 }
50             } catch (Exception e) {
51                 System.err.println(e.getMessage());
52             }
53
54             Collections.sort(numerosLeidos);
55
56             try (DataOutputStream ordenados = new DataOutputStream(new FileOutputStream(archivoOrdenados))) {
57                 for (int num : numerosLeidos) {
58                     ordenados.writeInt(num);
59                 }
60             } catch (Exception e) {
61                 System.err.println(e.getMessage());
62             }
63         }
64     }
```

```

64         System.out.println("Números ordenados guardados en el archivo: " + archivoOrdenados);
65
66     } catch (Exception e) {
67         System.err.println(e.getMessage());
68     }
69 }
70 }

```

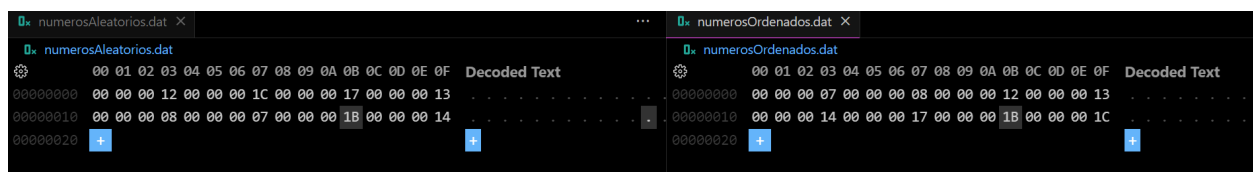
Consola:

```

Ingrese el nombre del archivo para guardar los números aleatorios: numerosAleatorios.dat
Ingrese el nombre del archivo para guardar los números ordenados: numerosOrdenados.dat
Ingrese el número de enteros (N): 8
Ingrese el valor mínimo del intervalo (x): 0
Ingrese el valor máximo del intervalo (y): 30
Números aleatorios guardados en el archivo: numerosAleatorios.dat
Números ordenados guardados en el archivo: numerosOrdenados.dat

```

Vista de archivos binarios con los números en hexadecimales:



Ejercicio 3:

Almacenar N objetos Alumno (cui, nombre, edad), leerlos, ordenarlos por edad y almacenarlos en otro archivo de objetos

Clase Alumno.java:

```

1  import java.io.*;
2
3  public class Alumno implements Serializable {
4      private String cui;
5      private String nombre;
6      private int edad;
7
8      public Alumno(String cui, String nombre, int edad) {
9          this.cui = cui;
10         this.nombre = nombre;
11         this.edad = edad;
12     }
13
14     public String getCui() {
15         return cui;
16     }

```

```

17
18     public String getNombre() {
19         return nombre;
20     }
21
22     public int getEdad() {
23         return edad;
24     }
25
26     @Override
27     public String toString() {
28         return "CUI: " + cui + ", Nombre: " + nombre + ", Edad: " + edad;
29     }
30 }
31

```

Clase Gestion.java:

```

1  import java.util.*;
2  import java.io.*;
3
4  public class Gestion {
5      public static void main(String[] args) {
6          Scanner sc = new Scanner(System.in);
7
8          try {
9              System.out.print("Ingrese el número de alumnos a registrar: ");
10             int n = sc.nextInt();
11             sc.nextLine();
12
13             List<Alumno> alumnos = new ArrayList<>();
14             for (int i = 0; i < n; i++) {
15                 System.out.println("Ingrese los datos del alumno " + (i + 1) + ":");
16                 System.out.print("CUI: ");
17                 String cui = sc.nextLine();
18                 System.out.print("Nombre: ");
19                 String nombre = sc.nextLine();
20                 System.out.print("Edad: ");
21                 int edad = sc.nextInt();
22                 sc.nextLine();
23
24                 alumnos.add(new Alumno(cui, nombre, edad));
25             }
26
27             try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("alumnos.bin"))) {
28                 for (Alumno a : alumnos) {
29                     out.writeObject(a);
30                 }
31             }
32
33             List<Alumno> alumnosLeidos = new ArrayList<>();
34             try (ObjectInputStream in = new ObjectInputStream(new FileInputStream("alumnos.bin"))) {
35                 while (true) {
36                     try {
37                         Alumno a = (Alumno) in.readObject();
38                         alumnosLeidos.add(a);
39                     } catch (EOFException e) {
40                         break;
41                     }
42                 }
43             }
44

```

```

45     System.out.println("\nAlumnos leídos del archivo:");
46     for (Alumno a : alumnosLeídos) {
47         System.out.println(a);
48     }
49
50     alumnosLeídos.sort(Comparator.comparingInt(Alumno::getEdad));
51
52     try (ObjectOutputStream out = new ObjectOutputStream(new FileOutputStream("alumnosOrdenados.bin"))) {
53         for (Alumno a : alumnosLeídos) {
54             out.writeObject(a);
55         }
56     }
57
58     System.out.println("\nAlumnos ordenados por edad y guardados en 'alumnosOrdenados.bin'.");
59 } catch (IOException | ClassNotFoundException e) {
60     System.err.println("Error: " + e.getMessage());
61 }
62 }
63 }

```

Consola:

```

Ingrese el número de alumnos a registrar: 3
Ingrese los datos del alumno 1:
CUI: 20232919
Nombre: Pepe
Edad: 20
Ingrese los datos del alumno 2:
CUI: 20198271
Nombre: Carlitos
Edad: 18
Ingrese los datos del alumno 3:
CUI: 20232188
Nombre: Jhonatan
Edad: 19

Alumnos leídos del archivo:
CUI: 20232919, Nombre: Pepe, Edad: 20
CUI: 20198271, Nombre: Carlitos, Edad: 18
CUI: 20232188, Nombre: Jhonatan, Edad: 19

```

Vista de la estructura del contenido de los archivos binarios:

alumnos.bin	alumnosOrdenados.bin
<pre> 00000000 AC ED 00 05 73 72 00 06 41 6C 75 6D 6E 6F 65 C0 ... s r ... Alumno e 00000010 DE 2F 32 20 F6 13 02 00 03 49 00 04 65 64 61 64 ... / 2 ... I ... edad 00000020 4C 00 03 63 75 69 74 00 12 4C 6A 61 76 61 2F 6C ... cu it ... L java / l 00000030 61 6E 67 2F 53 74 72 69 6E 67 3B 4C 00 06 6E 6F ... ang / String ; L ... no 00000040 6D 62 72 65 71 00 7E 00 01 78 70 00 00 00 14 74 ... mb req ... xp ... t 00000050 00 08 32 30 32 33 32 39 31 39 74 00 04 50 65 70 2 0 2 3 2 9 1 9 t ... Pep 00000060 65 73 71 00 7E 00 00 00 00 12 74 00 08 32 30 ... es q ... t ... 2 0 00000070 31 39 38 32 37 31 74 00 08 43 61 72 6C 69 74 6F ... 1 9 8 2 7 1 t ... Carlito 00000080 73 73 71 00 7E 00 00 00 00 13 74 00 08 32 30 ... ss q ... t ... 2 0 00000090 32 33 32 31 38 38 74 00 08 4A 68 6F 6E 61 74 61 ... 2 3 2 1 8 8 t ... Jhonata 000000A0 6E ... n </pre>	<pre> 00000000 AC ED 00 05 73 72 00 06 41 6C 75 6D 6E 6F 65 C0 ... s r ... Alumno e 00000010 DE 2F 32 20 F6 13 02 00 03 49 00 04 65 64 61 64 ... / 2 ... I ... edad 00000020 4C 00 03 63 75 69 74 00 12 4C 6A 61 76 61 2F 6C ... cu it ... L java / l 00000030 61 6E 67 2F 53 74 72 69 6E 67 3B 4C 00 06 6E 6F ... ang / String ; L ... no 00000040 6D 62 72 65 71 00 7E 00 01 78 70 00 00 00 12 74 ... mb req ... xp ... t 00000050 00 08 32 30 31 39 38 32 37 31 74 00 08 43 61 72 2 0 1 9 8 2 7 1 t ... Car 00000060 6C 69 74 6F 73 73 71 00 7E 00 00 00 00 13 74 ... litos sq ... t 00000070 00 08 32 30 32 33 32 31 38 38 74 00 08 4A 68 6F 2 0 2 3 2 1 8 8 t ... Jho 00000080 6E 61 74 61 6E 73 71 00 7E 00 00 00 00 14 74 ... natansq ... t 00000090 00 08 32 30 32 33 32 39 31 39 74 00 04 50 65 70 2 0 2 3 2 9 1 9 t ... Pep 000000A0 65 ... e </pre>