

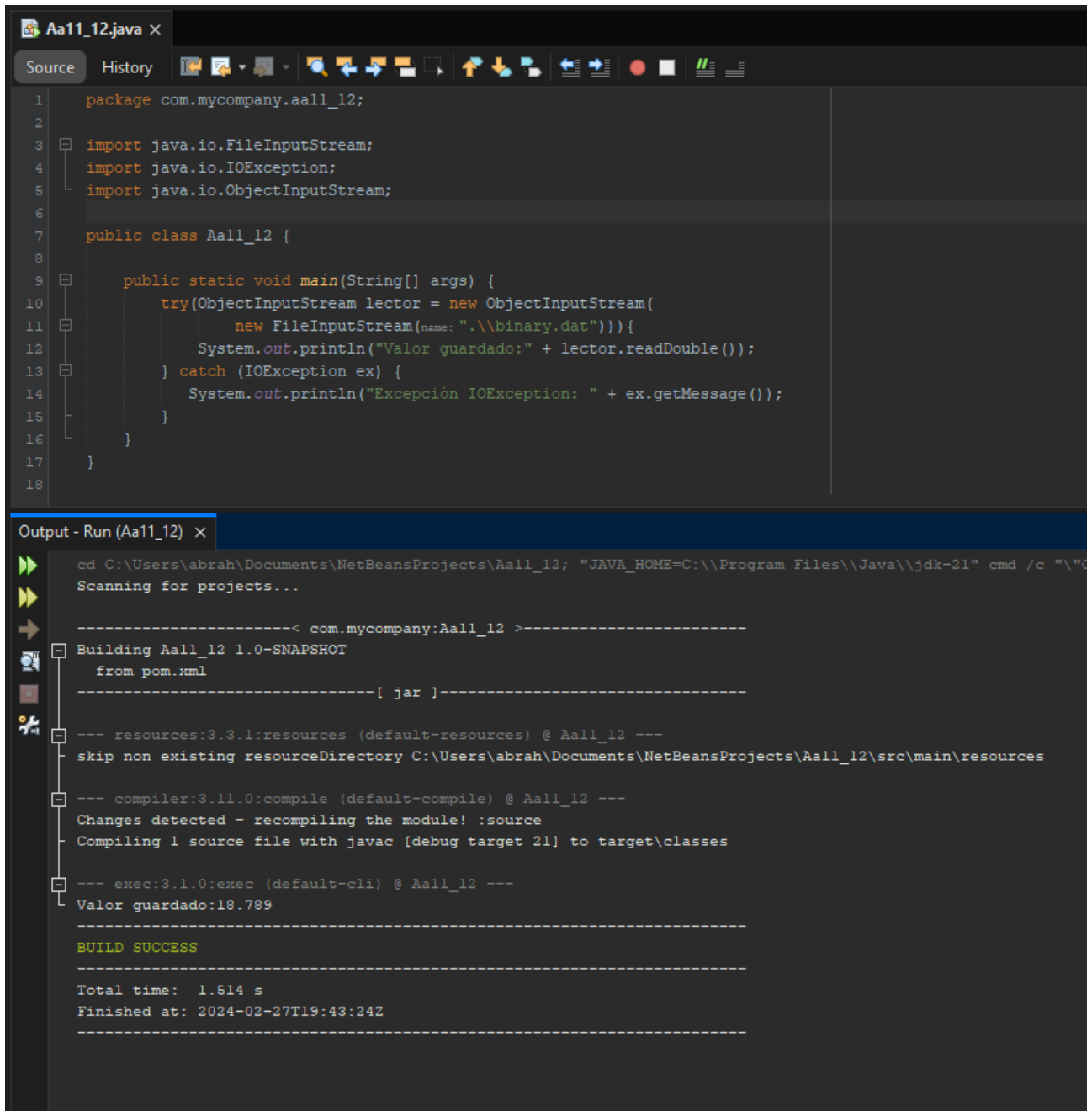
Actividades de aplicación. Realizar las siguientes 11.11, 11.12, 11.13, 11.14, 11.15, 11.16, 11.17, 11.18, 11.19 y 11.20.

**11.11.** Pide un valor `double` por consola y guárdalo en un archivo binario.

The screenshot displays an IDE environment with three main components:

- Source Editor:** Contains the Java code for `Aa11_11.java`. The code imports `FileNotFoundException`, `FileOutputStream`, `IOException`, `ObjectOutputStream`, and `Scanner`. The `main` method prompts the user to enter a `double` value, reads it from `System.in`, and writes it to a file named `binary.dat` using `ObjectOutputStream`. It includes catch blocks for `FileNotFoundException` and `IOException`.
- File Viewer:** A window titled `binary.dat` shows the raw binary data of the file, which appears as a sequence of characters: `~f DwB@2Éûçl<D`.
- Output - Run (Aa11\_11):** The terminal shows the execution of the program. It prompts for a `double` value, and the user enters `18,789`. The output indicates a successful build and execution, with a total time of 10.911 seconds and a finish time of 2024-02-27T19:37:36Z.

11.12. Abre el fichero de la Actividad de aplicación 11.11, lee el valor `double` contenido en él y muéstralo por pantalla.



The screenshot displays an IDE window with two tabs: 'Aa11\_12.java' and 'Output - Run (Aa11\_12)'. The 'Aa11\_12.java' tab shows the source code of a Java application. The code is as follows:

```
1 package com.mycompany.aall_12;
2
3 import java.io.FileInputStream;
4 import java.io.IOException;
5 import java.io.ObjectInputStream;
6
7 public class Aall_12 {
8
9     public static void main(String[] args) {
10         try(ObjectInputStream lector = new ObjectInputStream(
11             new FileInputStream("..\binary.dat"))){
12             System.out.println("Valor guardado:" + lector.readDouble());
13         } catch (IOException ex) {
14             System.out.println("Excepción IOException: " + ex.getMessage());
15         }
16     }
17 }
18
```

The 'Output - Run (Aa11\_12)' tab shows the execution output. The output includes the command used to run the application, the scanning process, the building of the application, the execution of the jar file, and the final output of the application. The output is as follows:

```
cd C:\Users\abrah\Documents\NetBeansProjects\Aall_12; "JAVA_HOME=C:\Program Files\Java\jdk-21" cmd /c ""
Scanning for projects...
-----< com.mycompany:Aall_12 >-----
Building Aall_12 1.0-SNAPSHOT
from pom.xml
-----[ jar ]-----
--- resources:3.3.1:resources (default-resources) @ Aall_12 ---
skip non existing resourceDirectory C:\Users\abrah\Documents\NetBeansProjects\Aall_12\src\main\resources
--- compiler:3.11.0:compile (default-compile) @ Aall_12 ---
Changes detected - recompiling the module! :source
Compiling 1 source file with javac [debug target 21] to target\classes
--- exec:3.1.0:exec (default-cli) @ Aall_12 ---
Valor guardado:18.789
BUILD SUCCESS
Total time: 1.514 s
Finished at: 2024-02-27T19:43:24Z
```

**11.13.** Escribe un programa que lea de un fichero binario una tabla de números `double` y después muestre el contenido de la tabla por consola.

```
1 package com.mycompany.aall_13;
2
3 import java.io.*;
4 import java.util.Arrays;
5
6 public class Aall_13 {
7
8     public static void main(String[] args) {
9         final double[] tabla = new double[]{12.3, 21.9, 89.74, 46.64, 22.15};
10        //Como no tengo ese fichero primero lo voy a crear
11        try (ObjectOutputStream salida = new ObjectOutputStream(
12            new FileOutputStream(name: ".\\numerosDouble.dat"))) {
13            salida.writeObject(obj:tabla);
14        } catch (IOException ex) {
15            System.out.println("Excepción IOException: " + ex.getMessage());
16        }
17        try (ObjectInputStream lector = new ObjectInputStream(
18            new FileInputStream(name: ".\\numerosDouble.dat"))) {
19            System.out.println("En el archivo numerosDouble.dat pone:\n "
20                + Arrays.toString( (double[])lector.readObject()));
21        } catch (IOException ex) {
22            System.out.println("Excepción IOException: " + ex.getMessage());
23        } catch (ClassNotFoundException ex) {
24            System.out.println("Excepción ClassNotFoundException: " + ex.getMessage());
25        }
26    }
27 }
28
```

Output - Run (Aa11\_13) x

```
--- resources:3.3.1:resources (default-resources) @ Aall_13 ---
skip non existing resourceDirectory C:\Users\abrah\Documents\NetBeansProjects\Aall_13\src\main\resources

--- compiler:3.11.0:compile (default-compile) @ Aall_13 ---
Changes detected - recompiling the module! :source
Compiling 1 source file with javac [debug target 21] to target\classes

--- exec:3.1.0:exec (default-cli) @ Aall_13 ---
En el archivo numerosDouble.dat pone:
[12.3, 21.9, 89.74, 46.64, 22.15]

-----
BUILD SUCCESS
-----

Total time: 1.480 s
Finished at: 2024-02-27T20:00:48Z
```

**11.14.** Introduce por teclado una frase y guárdala en un archivo binario. A continuación, recuérrela y muéstrala por pantalla.

```
1 package com.mycompany.aall_14;
2
3 import java.io.*;
4 import java.util.Scanner;
5
6 public class Aall_14 {
7
8     public static void main(String[] args) {
9         Scanner sc = new Scanner(System.in);
10        try (ObjectOutputStream salida = new ObjectOutputStream(
11            new FileOutputStream(name: ".\\frase.dat"))) {
12            System.out.println(x: "Introduzca frase a almacenar en binario:");
13            salida.writeObject(obj: sc.nextLine());
14        } catch (IOException ex) {
15            System.out.println("Excepción IOException: " + ex.getMessage());
16        }
17        try (ObjectInputStream lector = new ObjectInputStream(
18            new FileInputStream(name: ".\\frase.dat"))) {
19            System.out.println("En el archivo frase.dat pone:\n "
20                + (String) lector.readObject());
21        } catch (IOException ex) {
22            System.out.println("Excepción IOException: " + ex.getMessage());
23        } catch (ClassNotFoundException ex) {
24            System.out.println("Excepción ClassNotFoundException: " + ex.getMessage());
25        }
26    }
27 }
28
```

Output - Run (Aa11\_14) x

```
skip non existing resourceDirectory C:\Users\abrah\Documents\NetBeansProjects\Aa11_14\src\ma
--- compiler:3.11.0:compile (default-compile) @ Aa11_14 ---
Changes detected - recompiling the module! :source
Compiling 1 source file with javac [debug target 21] to target\classes
--- exec:3.11.0:exec (default-cli) @ Aa11_14 ---
Introduzca frase a almacenar en binario:
Hola me llamo Abraham Quintana y esta es la frase a guardar
En el archivo frase.dat pone:
Hola me llamo Abraham Quintana y esta es la frase a guardar
-----
BUILD SUCCESS
-----
Total time: 18.480 s
Finished at: 2024-02-27T20:06:14Z
```

**11.15.** Implementa un programa que lea números enteros desde el fichero *numeros.dat* y los vaya guardando en los ficheros *pares.dat* e *impares.dat*, según su paridad.

The screenshot shows an IDE with a project explorer on the left, a code editor in the center, and an output console at the bottom.

**Project Explorer:** The left sidebar shows a project structure with folders Aa10\_17 through Aa10\_23, Aa11\_11 (selected), Aa11\_12 through Aa11\_15, a 'src' folder containing 'impares.dat', 'numeros.dat', and 'pares.dat', a 'target' folder, and a 'pom.xml' file.

**Code Editor:** The center pane displays Java code for a class named `Aa11_15`. The code defines a `main` method that reads integers from `numeros.dat` and writes them to `pares.dat` or `impares.dat` based on their parity. The code includes imports for `java.io.*` and uses `ObjectOutputStream` and `ObjectInputStream` for file operations.

```
1 package com.mycompany.a11_15;
2
3 import java.io.*;
4
5 public class Aa11_15 {
6
7     public static void main(String[] args) {
8         int[] enteros = new int[]{39, 47, 5, 99, 854, 34};
9         int numero;
10        try (ObjectOutputStream salidaNumeros = new ObjectOutputStream(new FileOutputStream(name: ".\\numeros.dat"));
11             ObjectOutputStream salidaPares = new ObjectOutputStream(new FileOutputStream(name: ".\\pares.dat"));
12             ObjectOutputStream salidaImpares = new ObjectOutputStream(new FileOutputStream(name: ".\\impares.dat")))) {
13            for (int num : enteros) { //inserto los numeros 1 a 1 para que no sea una tabla.
14                salidaNumeros.writeInt(num);
15            }
16
17            try (ObjectInputStream lector = new ObjectInputStream(new FileInputStream(name: ".\\numeros.dat"))) {
18                while (true) {
19                    numero = lector.readInt();
20                    if (numero % 2 == 0) {
21                        salidaPares.writeInt(numero);
22                    } else {
23                        salidaImpares.writeInt(numero);
24                    }
25                }
26            } catch (EOFException ex) {
27                System.out.println("Fin del fichero");
28            }
29        } catch (IOException ex) {
30            System.out.println("Excepción IOException: " + ex);
31        }
32    }
33 }
34
```

**Output Console:** The bottom pane shows the execution output. It includes messages from the compiler, IDE changes, and the execution of the program. The output shows "Fin del fichero" and "BUILD SUCCESS".

```
Output X
Debugger Console X Run (Aa11_15) X
--- Compiler:3.11.0:compile (default-compile) @ Aa11_15 ---
Changes detected - recompiling the module! :source
Compiling 1 source file with javac [debug target 21] to target\classes
--- exec:3.11.0:exec (default-cli) @ Aa11_15 ---
Fin del fichero
BUILD SUCCESS
Total time: 1.246 s
```

**11.16.** Implementa una aplicación que gestione una lista de nombres ordenada por orden alfabético. Al arrancar se leerá de un fichero los nombres insertados anteriormente y se pedirán nombres nuevos hasta que se introduzca la cadena «fin». Cada nombre que se introduzca deberá añadirse a los que ya había, de forma que la lista permanezca ordenada. Al terminar, se guardará en el fichero la lista actualizada.

```

1  package com.mycompany.aall_16;
2
3  import java.io.*;
4  import java.util.Arrays;
5  import java.util.Scanner;
6
7  public class Aall_16 {
8
9      public static void main(String[] args) {
10         String[] nombres = new String[]{"Abraham", "Benearo", "Lucas", "Maria"};
11         String[] aux;
12         String nombre;
13         Scanner sc = new Scanner(System.in);
14         try (ObjectOutputStream escritor = new ObjectOutputStream(new FileOutputStream(name: ".\\nombres.dat"));
15              ObjectInputStream lector = new ObjectInputStream(new FileInputStream(name: ".\\nombres.dat"))) {
16             escritor.writeObject(obj:nombres); //primero creo el fichero para tenerlo
17             nombres = (String[]) lector.readObject();
18             do{
19                 System.out.println(x: "Introduzca siguiente nombre o fin para terminar:");
20                 nombre = sc.nextLine();
21                 if(!nombre.equals(anObject: "fin")){
22                     aux = new String[nombres.length + 1];
23                     System.arraycopy(src:nombres, srcPos:0, dest: aux, destPos:0, length:nombres.length);
24                     aux[nombres.length] = nombre;
25                     Arrays.sort(a: aux);
26                     nombres = aux;
27                 }
28             }while(!nombre.equals(anObject: "fin"));
29             escritor.writeObject(obj:nombres);
30             System.out.println("Resultado:\n" + Arrays.toString((String[])lector.readObject()));
31         } catch (FileNotFoundException ex) {
32             System.out.println("Excepción FileNotFoundException: " + ex);
33         } catch (IOException ex) {
34             System.out.println("Excepción IOException: " + ex);
35         } catch (ClassNotFoundException ex) {
36             System.out.println("Excepción ClassNotFoundException: " + ex);
37         }
38     }
39 }

```

```

--- exec:3.1.0:exec (default-cli) @ Aall_16 ---
Introduzca siguiente nombre o fin para terminar:
Francisco
Introduzca siguiente nombre o fin para terminar:
Marta
Introduzca siguiente nombre o fin para terminar:
Antonio
Introduzca siguiente nombre o fin para terminar:
fin
Resultado:
[Abraham, Antonio, Benearo, Francisco, Lucas, Maria, Marta]
-----
BUILD SUCCESS
-----
Total time: 37.159 s

```

**11.17.** Escribe un texto, línea a línea, de forma que, cada vez que se pulse Intro, se guarde la línea en un archivo binario. El proceso se termina cuando introduzcamos una línea vacía. Después el programa lee el texto completo del archivo y lo muestra por pantalla.

```
1 package com.mycompany.aall_17;
2
3 import java.io.*;
4 import java.util.Scanner;
5
6 public class Aall_17 {
7
8     public static void main(String[] args) {
9         Scanner sc = new Scanner(System.in);
10        String linea;
11        try (ObjectOutputStream escritor = new ObjectOutputStream(new FileOutputStream(name: ".\\texto.dat"));
12             ObjectInputStream lector = new ObjectInputStream(new FileInputStream(name: ".\\texto.dat"))) {
13            System.out.println("Introduzca texto a guardar o pulse ENTER para salir:");
14            do{
15                linea = sc.nextLine();
16                if(!linea.equals("")){
17                    escritor.writeObject(linea);
18                }
19            }while(!linea.equals(""));
20
21            System.out.println("Resultado:\n");
22            while(true){
23                System.out.println((String)lector.readObject());
24            }
25        }catch (EOFException e){
26            System.out.println("*****Fin del fichero*****");
27        } catch (FileNotFoundException ex) {
28            System.out.println("Excepción FileNotFoundException: " + ex);
29        } catch (IOException ex) {
30            System.out.println("Excepción IOException: " + ex);
31        } catch (ClassNotFoundException ex) {
32            System.out.println("Excepción ClassNotFoundException: " + ex);
33        }
34    }
35 }
```

```
--- exec:3.1.0:exec (default-cli) @ Aall_17 ---
Introduzca texto a guardar o pulse ENTER para salir:
hola que tal
este es el texto a guardar
soy pepito
y termino con esta

Resultado:

hola que tal
este es el texto a guardar
soy pepito
y termino con esta
*****Fin del fichero*****
-----
BUILD SUCCESS
-----
```

**11.18.** Un libro de firmas es útil para recoger los nombres de todas las personas que han pasado por un determinado lugar. Crea una aplicación que permita mostrar el libro de firmas o insertar un nuevo nombre (comprobando que no se encuentre repetido) usando el fichero binario *firmas.dat*.

```
package com.mycompany.aall_18;

import java.io.*;
import java.util.Scanner;

public class Aall_18 {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String firma;

        try (ObjectOutputStream escritor = new ObjectOutputStream(new FileOutputStream(name: ".\\firmas.dat"))) {
            do {
                System.out.println("Introduzca firma a guardar, pulse 1 "
                    + "para mostrar el contenido del libro y ENTER para salir:");
                firma = sc.nextLine();
                switch (firma) {
                    case ("") -> {
                    }
                    case ("1") -> {
                        try (ObjectInputStream lector = new ObjectInputStream(new FileInputStream(name: ".\\firmas.dat"))) {
                            System.out.println(x: "Resultado:\n");
                            while (true) {
                                System.out.println((String)lector.readObject());
                            }
                        } catch (EOFException e) {
                            System.out.println(x: "*****Fin del fichero*****");
                        }
                    }
                    default -> {
                        escritor.writeObject(obj:firma);
                        escritor.flush();
                    }
                }
            } while (!firma.equals(obj:""));
        } catch (FileNotFoundException ex) {
            System.out.println("Excepción FileNotFoundException: " + ex);
        } catch (IOException ex) {
            System.out.println("Excepción IOException: " + ex);
        } catch (ClassNotFoundException ex) {
            System.out.println("Excepción ClassNotFoundException: " + ex);
        }
    }
}
```

```
--- exec:3.1.0:exec (default-cli) @ Aall_18 ---
Introduzca firma a guardar, pulse 1 para mostrar el contenido del libro y ENTER para salir:
Abraham
Introduzca firma a guardar, pulse 1 para mostrar el contenido del libro y ENTER para salir:
Tu
Introduzca firma a guardar, pulse 1 para mostrar el contenido del libro y ENTER para salir:
Yo
Introduzca firma a guardar, pulse 1 para mostrar el contenido del libro y ENTER para salir:
1
Resultado:

Abraham
Tu
Yo
*****Fin del fichero*****
Introduzca firma a guardar, pulse 1 para mostrar el contenido del libro y ENTER para salir:
Paco
Introduzca firma a guardar, pulse 1 para mostrar el contenido del libro y ENTER para salir:
Pedro
Introduzca firma a guardar, pulse 1 para mostrar el contenido del libro y ENTER para salir:
1
Resultado:

Abraham
Tu
Yo
Paco
Pedro
*****Fin del fichero*****
Introduzca firma a guardar, pulse 1 para mostrar el contenido del libro y ENTER para salir:

-----
BUILD SUCCESS
-----
Bash: 08-055
```



**11.19.** Por motivos puramente estadísticos se desea llevar constancia del número de llamadas recibidas cada día en una oficina. Para ello, al terminar cada jornada laboral se guarda dicho número al final de un archivo binario. Implementa una aplicación con un menú, que nos permita añadir el número correspondiente cada día y ver la lista completa en cualquier momento.

```

1 package com.myccompany.aall_19;
2
3 import java.io.*;
4 import java.util.Scanner;
5
6 public class Aall_19 {
7     public static void main(String[] args) {
8         Scanner sc = new Scanner(System.in);
9         String dato;
10        Registro registro;
11
12        try (ObjectOutputStream escritor = new ObjectOutputStream(new FileOutputStream(name: ".\\llamadas.dat"))) {
13            do {
14                System.out.println("Introduzca fecha(dd/mm/aaaa), pulse ver "
15                    + "para mostrar el listado o fin para salir:");
16                dato = sc.next();
17                switch(dato) {
18                    case ("fin") -> {
19                        }
20                    case ("ver") -> {
21                        try (ObjectInputStream lector = new ObjectInputStream(new FileInputStream(name: ".\\llamadas.dat"))) {
22                            System.out.println("Llamadas por día:\n");
23                            while (true) {
24                                System.out.println((Registro) lector.readObject());
25                            }
26                        } catch (EOFException e) {
27                            System.out.println("*****Fin del fichero*****");
28                        }
29                    }
30                    default -> {
31                        registro = new Registro();
32                        registro.dia = dato;
33                        System.out.println("Introduzca llamadas de hoy: ");
34                        registro.llamadas = sc.nextInt();
35                        escritor.writeObject(registro);
36                        escritor.flush();
37                    }
38                }
39            } while (!dato.equals("fin"));
40        } catch (FileNotFoundException ex) {
41            System.out.println("Excepción FileNotFoundException: " + ex);
42        } catch (IOException ex) {
43            System.out.println("Excepción IOException: " + ex);
44        } catch (ClassNotFoundException ex) {
45            System.out.println("Excepción ClassNotFoundException: " + ex);
46        }
47    }
48
49    private static class Registro implements Serializable{
50        String dia; //podría haber utilizado Date, pero no creo que sea la finalidad del ejercicio
51        int llamadas;
52
53        @Override
54        public String toString() {
55            return "Registro[" + "d\u00eda=" + dia + ", llamadas=" + llamadas + "]";
56        }
57    }
58 }
59
60

```

```

--- exec:3.1.0:exec (default-cli) @ Aall_19 ---
Introduzca fecha(dd/mm/aaaa), pulse ver para mostrar el listado o fin para salir:
01/01/2020
Introduzca llamadas de hoy:
45
Introduzca fecha(dd/mm/aaaa), pulse ver para mostrar el listado o fin para salir:
02/05/2020
Introduzca llamadas de hoy:
64
Introduzca fecha(dd/mm/aaaa), pulse ver para mostrar el listado o fin para salir:
09/05/2021
Introduzca llamadas de hoy:
14
Introduzca fecha(dd/mm/aaaa), pulse ver para mostrar el listado o fin para salir:
ver
llamadas por día:

Registro{dia=01/01/2020, llamadas=45}
Registro{dia=02/05/2020, llamadas=64}
Registro{dia=09/05/2021, llamadas=14}
*****Fin del fichero*****
Introduzca fecha(dd/mm/aaaa), pulse ver para mostrar el listado o fin para salir:
fin

BUILD SUCCESS

Total time: 50.202 s

```

**11.20.** Implementa una aplicación que permita guardar y recuperar los datos de los clientes de una empresa. Para ello, define la clase `Cliente`, que tendrá los atributos: `id` (identificador de cliente), `nombre` y `telefono`. Los objetos `Cliente` se insertarán en una tabla.

Para realizar las distintas operaciones, la aplicación tendrá el siguiente menú:

1. Añadir nuevo cliente.
2. Modificar datos.
3. Dar de baja cliente.
4. Listar los clientes.

La información se guardará en un fichero binario, que se cargará en la memoria al iniciar la aplicación y se grabará en disco, actualizada, al terminar.

```

1 package com.myccompany.sail_20;
2
3 import java.io.*;
4 import java.util.Arrays;
5 import java.util.Scanner;
6
7 public class Sail_20 {
8
9     public static void main(String[] args) {
10         Cliente[] clientes = new Cliente[0];
11         Cliente[] aux;
12         Cliente cliente;
13         String id, nombre, telefono;
14         int opcion;
15         Scanner sc = new Scanner(System.in);
16         try (ObjectInputStream lector = new ObjectInputStream(
17             new FileInputStream(name: ".\\clientes.dat"))) {
18             while (true) {
19                 clientes = (Cliente[]) lector.readObject();
20             }
21         } catch (EOFException e) {
22         } catch (Exception e) {
23             System.out.println("Ha ocurrido una excepción: " + e);
24         }
25         do {
26             System.out.println("
27
28             *****Menú de Clientes*****
29             1 Añadir cliente
30             2 Modificar datos
31             3 Dar de baja cliente
32             4 Listar clientes
33             0 Salir:");
34             opcion = sc.nextInt();
35             sc.skip(pattern: "\n");
36             switch (opcion) {
37                 case (1) -> { //Añadir
38                     System.out.println("Introduzca id (no se podrá modificar):");
39                     id = sc.nextLine();
40                     System.out.println("Nombre:");
41                     nombre = sc.nextLine();
42                     System.out.println("Teléfono:");
43                     telefono = sc.nextLine();
44                     cliente = new Cliente(id, nombre, telefono);
45                     aux = new Cliente[clientes.length + 1];
46                     System.arraycopy(src: clientes, srcPos: 0, dest: aux, destPos: 0, length: clientes.length);
47                     aux[clientes.length] = cliente;
48                     clientes = aux;
49                     aux = null;
50                     try (ObjectOutputStream escritor = new ObjectOutputStream(
51                         new FileOutputStream(name: ".\\clientes.dat"))) {
52                         escritor.writeObject(obj: clientes);
53                         escritor.flush();
54                     } catch (Exception e) {
55                         System.out.println("Ha ocurrido una excepción: " + e);
56                     }
57                 }
58                 case (2) -> { //Modificar
59                     System.out.println("Introduzca id cliente a modificar:");
60                     id = sc.nextLine();
61                     for (Cliente c : clientes) {
62                         if (c.getId().equals(id)) {
63                             sc.skip(pattern: "\n");
64                             System.out.println("Cliente encontrado\n
65                             + "Introduzca nuevo nombre:");
66                             c.setNombre(nombre: sc.nextLine());
67                             System.out.println("Introduzca nuevo teléfono:");
68                             c.setTelefono(telefono: sc.nextLine());
69                             try (ObjectOutputStream escritor = new ObjectOutputStream(
70                                 new FileOutputStream(name: ".\\clientes.dat"))) {
71                                 escritor.writeObject(obj: clientes);
72                                 escritor.flush();
73                             } catch (Exception e) {
74                                 System.out.println("Ha ocurrido una excepción: " + e);
75                             }
76                             break;
77                         }
78                     }
79                 }
80                 case (3) -> { //dar de baja
81                     System.out.println("Introduzca id cliente a dar de baja:");
82                     id = sc.nextLine();
83                     for (int i = 0; i < clientes.length; i++) {
84                         if (clientes[i].getId().equals(id)) {
85                             aux = new Cliente[clientes.length - 1];
86                             System.arraycopy(src: clientes, srcPos: 0, dest: aux,
87                                 destPos: 0, length: i);
88                             System.arraycopy(src: clientes, srcPos: i + 1, dest: aux,
89                                 destPos: i, length: clientes.length - i - 1);
90                             clientes = aux;
91                             aux = null;
92                         }
93                     }
94                 }
95             }
96         } while (opcion != 0);
97     }
98 }

```

```

1 package com.myccompany.sail_20;
2
3 import java.io.*;
4
5 public class Cliente implements Serializable {
6     private String id, nombre, telefono;
7
8     public Cliente(String id, String nombre, String telefono) {
9         this.id = id;
10        this.nombre = nombre;
11        this.telefono = telefono;
12    }
13
14    public String getId() {
15        return id;
16    }
17
18    //Protejo el id para que no se pueda modificar una vez creado
19    private void setId(String id) {
20        this.id = id;
21    }
22
23    public String getNombre() {
24        return nombre;
25    }
26
27    public void setNombre(String nombre) {
28        this.nombre = nombre;
29    }
30
31    public String getTelefono() {
32        return telefono;
33    }
34
35    public void setTelefono(String telefono) {
36        this.telefono = telefono;
37    }
38
39    @Override
40    public String toString() {
41        return "\n***Cliente***: " + "\n-id=" + id + "\n-nombre=" + nombre
42            + "\n-telefono=" + telefono + "\n";
43    }
44 }
45
46

```

```

91         try (ObjectOutputStream escritor = new ObjectOutputStream(
92             new FileOutputStream(name: ".\\clientes.dat"))) {
93             escritor.writeObject(obj.clientes);
94             escritor.flush();
95         } catch (Exception e) {
96             System.out.println("Ha ocurrido una excepción: " + e);
97         }
98         System.out.println(s: "\nCliente eliminado\n");
99         break;
100     }
101 }
102 }
103 case (4) -> { //Listar
104     try (ObjectInputStream lector = new ObjectInputStream(
105         new FileInputStream(name: ".\\clientes.dat"))) {
106         while (true) {
107             System.out.println(s: Arrays.toString(
108                 (Cliente[]) lector.readObject()));
109         }
110     } catch (EOFException e) {
111         System.out.println("*****Fin del directorio"
112             + "*****");
113     } catch (Exception e) {
114         System.out.println("Ha ocurrido una excepción: " + e);
115     }
116 }
117 case (0) -> {
118 }
119 default -> {
120     System.out.println(s: "Opción errónea");
121 }
122 } while (opcion != 0);
123 }
124 }
125 }
126 }

```

1 Listo clientes, añado 222b, listo clientes

2 modifico 222b, listo clientes, elimino 222b

```

--- exec:3.1.0:exec (default-cli) @ Aall_20 ---
*****Menú Clientes*****
1 Añadir cliente
2 Modificar datos
3 Dar de baja cliente
4 Listar clientes
0 Salir:
4
[
***Cliente***:
-id=111a
-nombre=Abraham Quintana
-telefono=111 111 111
]
*****Fin del directorio*****
*****Menú Clientes*****
1 Añadir cliente
2 Modificar datos
3 Dar de baja cliente
4 Listar clientes
0 Salir:
1
Introduzca id (no se podrá modificar):
222b
Nombre:
Pepé Lucas
Teléfono:
333 333 333
*****Menú Clientes*****
1 Añadir cliente
2 Modificar datos
3 Dar de baja cliente
4 Listar clientes
0 Salir:
4
[
***Cliente***:
-id=111a
-nombre=Abraham Quintana
-telefono=111 111 111
,
***Cliente***:
-id=222b
-nombre=Pepé Lucas
-telefono=333 333 333
]
*****Fin del directorio*****

```

```

*****Menú Clientes*****
1 Añadir cliente
2 Modificar datos
3 Dar de baja cliente
4 Listar clientes
0 Salir:
2
Introduzca id cliente a modificar;
222b
Cliente encontrado
Introduzca nuevo nombre:
Elon Mask
Introduzca nuevo teléfono:
999 999 999
*****Menú Clientes*****
1 Añadir cliente
2 Modificar datos
3 Dar de baja cliente
4 Listar clientes
0 Salir:
4
[
***Cliente***:
-id=111a
-nombre=Abraham Quintana
-telefono=111 111 111
,
***Cliente***:
-id=222b
-nombre=Elon Mask
-telefono=999 999 999
]
*****Fin del directorio*****
*****Menú Clientes*****
1 Añadir cliente
2 Modificar datos
3 Dar de baja cliente
4 Listar clientes
0 Salir:
3
Introduzca id cliente a dar de baja;
222b

Cliente eliminado

*****Menú Clientes*****
1 Añadir cliente
2 Modificar datos

```

### 3 Listo clientes y salgo.

```
*****Menú Clientes*****
1 Añadir cliente
2 Modificar datos
3 Dar de baja cliente
4 Listar clientes
0 Salir:
4
[
***Cliente***:
-id=111a
-nombre=Abraham Quintana
-telefono=111 111 111
]
*****Fin del directorio*****
*****Menú Clientes*****
1 Añadir cliente
2 Modificar datos
3 Dar de baja cliente
4 Listar clientes
0 Salir:
0
-----
BUILD SUCCESS
-----
Total time: 01:03 min
Finished at: 2024-03-01T21:26:22Z
```