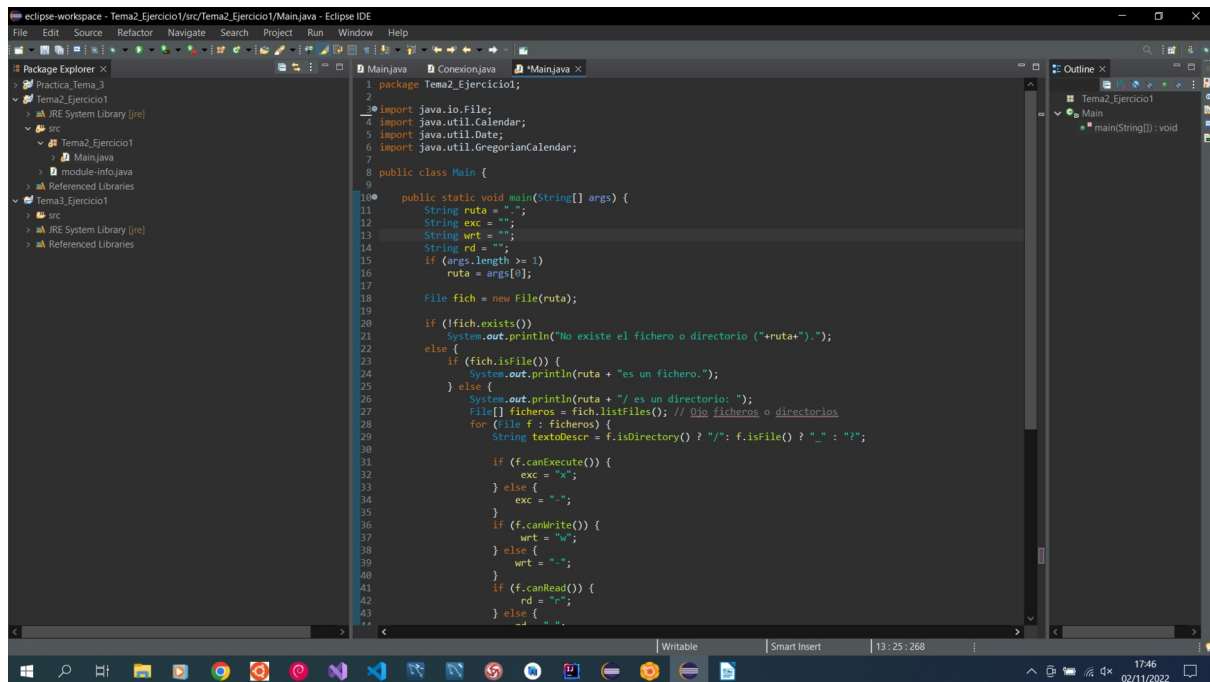


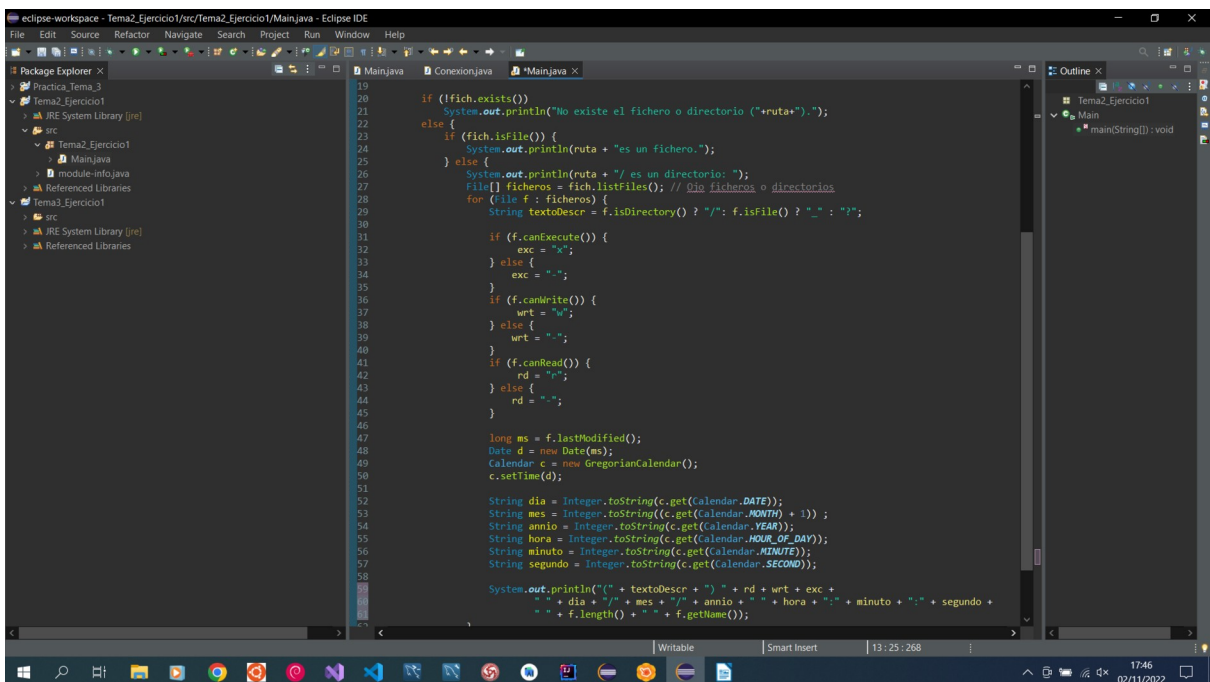
# Ejercicios Tema 2: Ficheros

## 2.3 – LA CLASE FILE EN JAVA

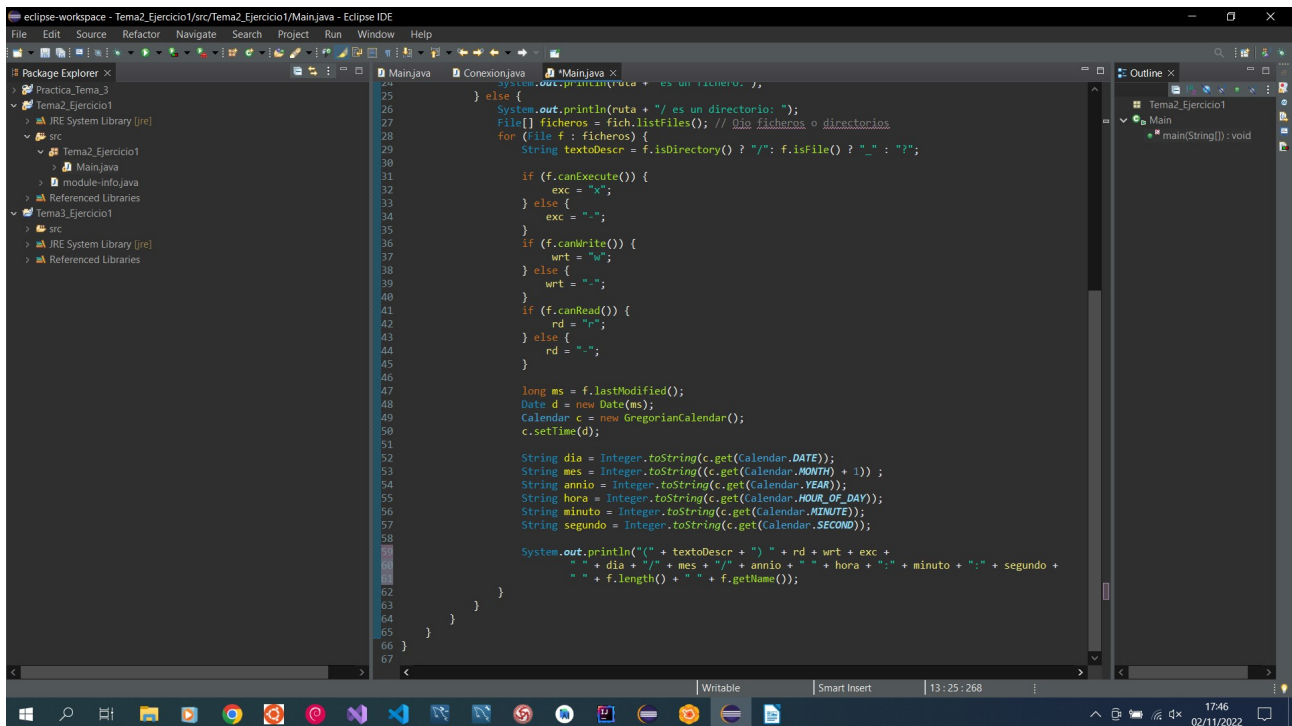
a) Modifica el programa anterior para que muestre más información acerca de cada fichero y directorio, al menos el tamaño (si es un fichero), los permisos de los que se dispone sobre el fichero o directorio, y la fecha de última modificación. Los permisos hay que mostrarlos en el formato Linux (rwx ó r-x).



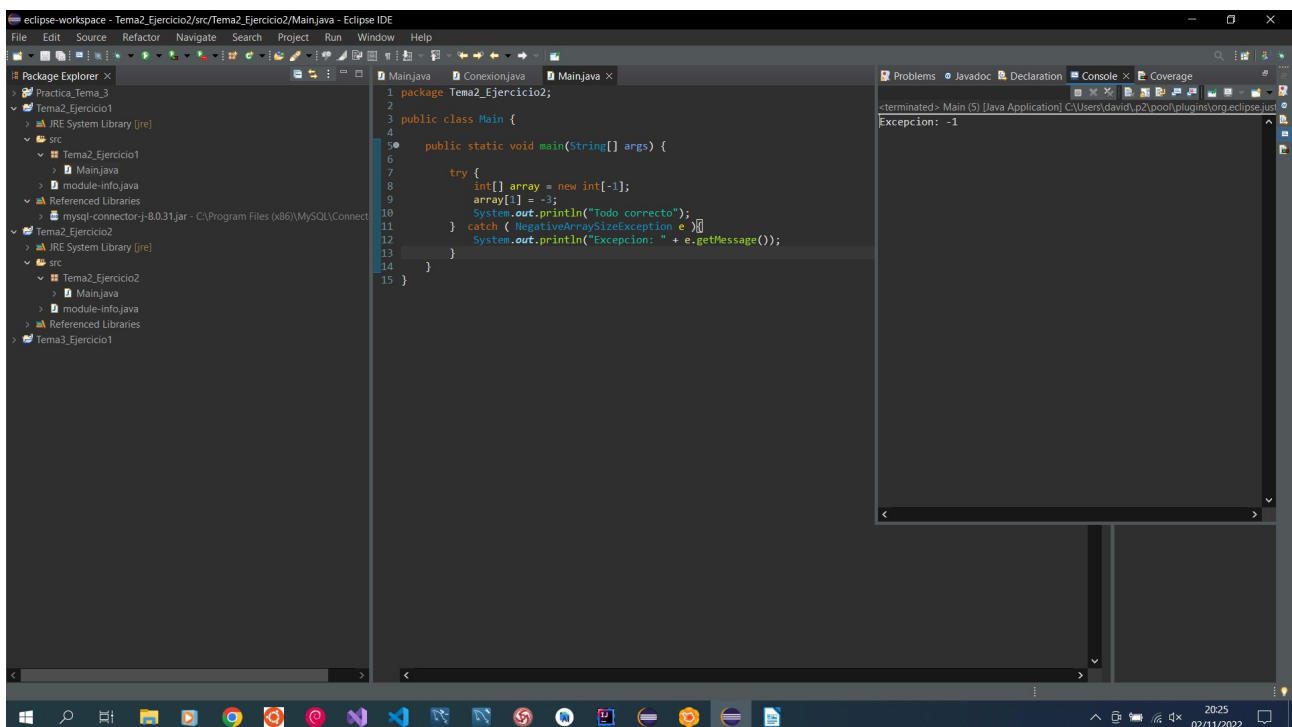
```
1 package Tema2_Ejercicio1;
2
3 import java.io.File;
4 import java.util.Calendar;
5 import java.util.Date;
6 import java.util.GregorianCalendar;
7
8 public class Main {
9
10     public static void main(String[] args) {
11         String ruta = ".";
12         String exc = "";
13         String wrt = "";
14         String rd = "";
15         if (args.length >= 1)
16             ruta = args[0];
17
18         File fich = new File(ruta);
19
20         if (!fich.exists())
21             System.out.println("No existe el fichero o directorio (" + ruta + ").");
22         else {
23             if (fich.isFile()) {
24                 System.out.println(ruta + "es un fichero.");
25             } else {
26                 System.out.println(ruta + "/" + "es un directorio.");
27                 File[] ficheros = fich.listFiles(); // Día ficheros o directorios
28                 for (File f : ficheros) {
29                     String textoDescr = f.isDirectory() ? "/" : f.isFile() ? "." : "?";
30
31                     if (f.canExecute()) {
32                         exc = "x";
33                     } else {
34                         exc = "-";
35                     }
36                     if (f.canWrite()) {
37                         wrt = "w";
38                     } else {
39                         wrt = "-";
40                     }
41                     if (f.canRead()) {
42                         rd = "r";
43                     } else {
44                         rd = "-";
45                     }
46                 }
47             }
48         }
49     }
50 }
```



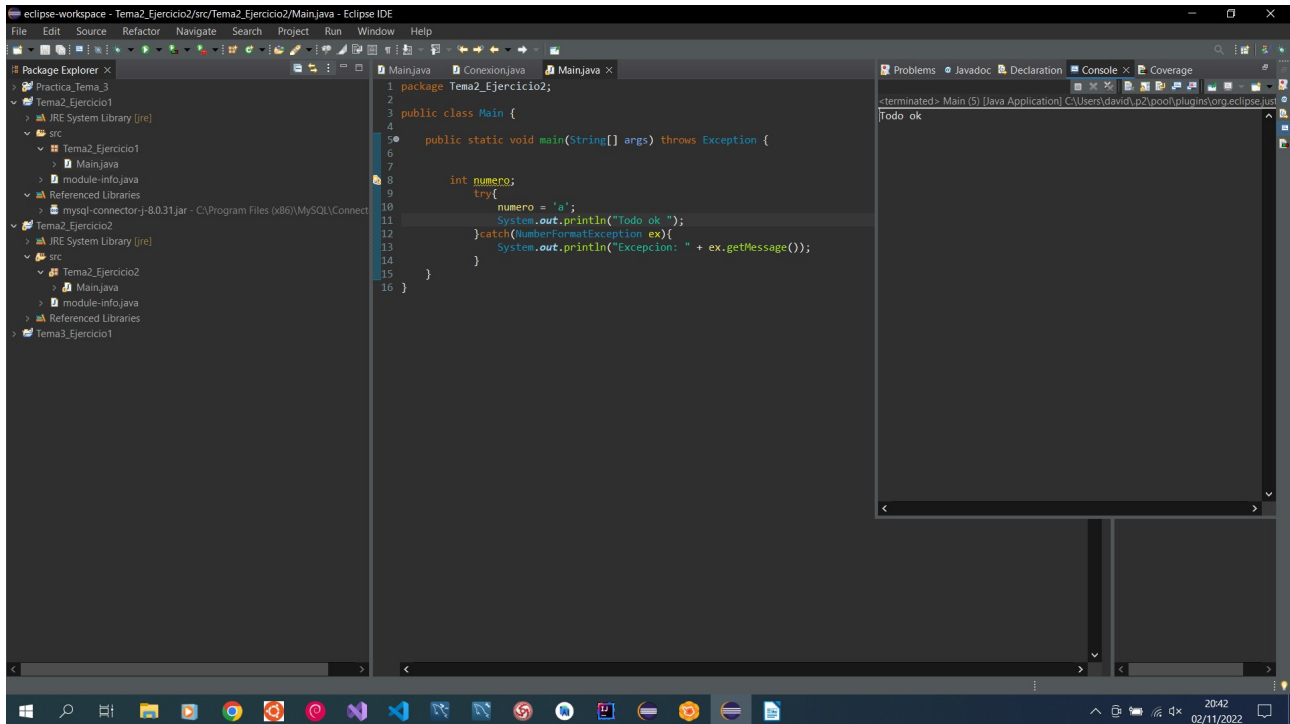
```
19
20 if (!fich.exists())
21     System.out.println("No existe el fichero o directorio (" + ruta + ").");
22 else {
23     if (fich.isFile()) {
24         System.out.println(ruta + "es un fichero.");
25     } else {
26         System.out.println(ruta + "/" + "es un directorio.");
27         File[] ficheros = fich.listFiles(); // Día ficheros o directorios
28         for (File f : ficheros) {
29             String textoDescr = f.isDirectory() ? "/" : f.isFile() ? "." : "?";
30
31             if (f.canExecute()) {
32                 exc = "x";
33             } else {
34                 exc = "-";
35             }
36             if (f.canWrite()) {
37                 wrt = "w";
38             } else {
39                 wrt = "-";
40             }
41             if (f.canRead()) {
42                 rd = "r";
43             } else {
44                 rd = "-";
45             }
46
47             long ms = f.lastModified();
48             Date d = new Date(ms);
49             Calendar c = new GregorianCalendar();
50             c.setTime(d);
51
52             String dia = Integer.toString(c.get(Calendar.DATE));
53             String mes = Integer.toString(c.get(Calendar.MONTH) + 1);
54             String anno = Integer.toString(c.get(Calendar.YEAR));
55             String hora = Integer.toString(c.get(Calendar.HOUR_OF_DAY));
56             String minuto = Integer.toString(c.get(Calendar.MINUTE));
57             String segundo = Integer.toString(c.get(Calendar.SECOND));
58
59             System.out.println("(" + textoDescr + " " + rd + wrt + exc +
60                 " " + dia + "/" + mes + "/" + anno + " " + hora + ":" + minuto + ":" + segundo +
61                 " " + f.length() + " " + f.getName());
62         }
63     }
64 }
```



a) Crea un programa en Java que gestione excepciones de la siguiente forma: Crea un array y asigna a un elemento el valor -3. Comprueba la excepción generada y usa un bloque try-catch para controlar el error.

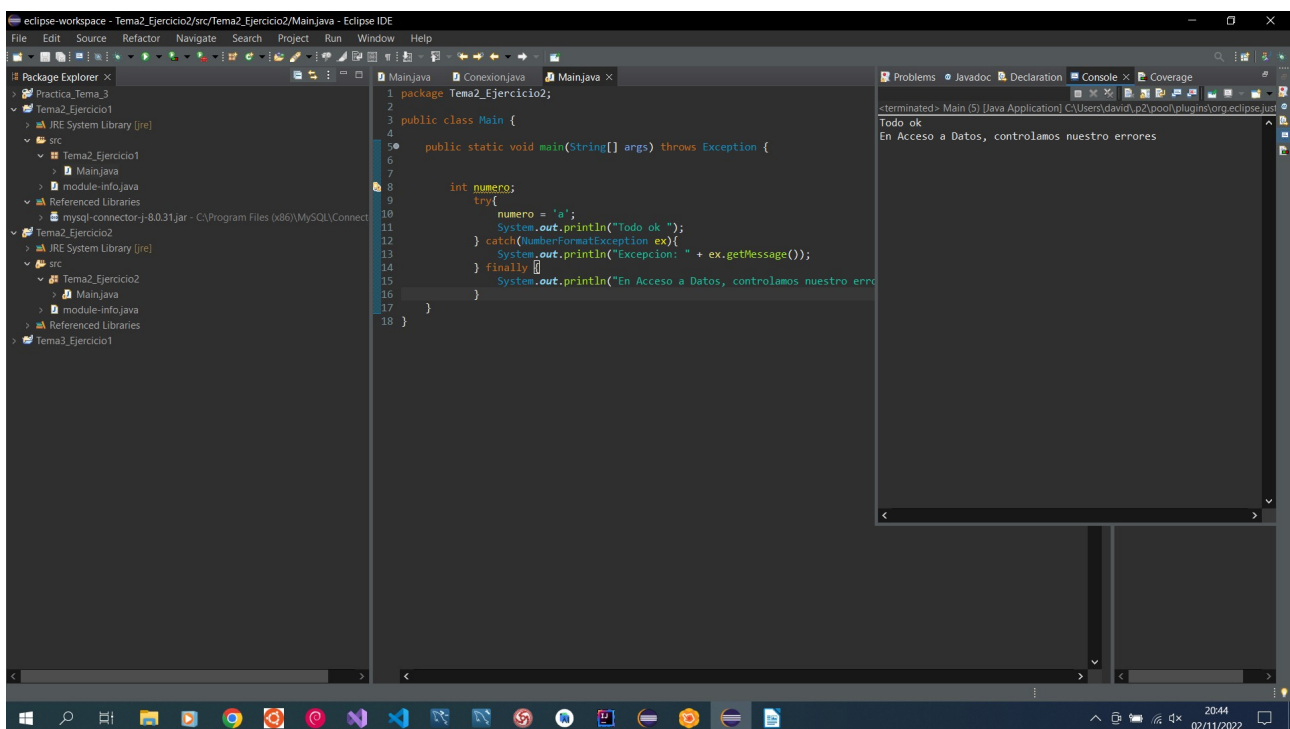


b) Modifica el programa añadiendo la creación de un entero y asignándole un valor de tipo carácter. Comprueba la nueva excepción y después incluye el bloque try-catch el control de la nueva excepción.



```
1 package Tema2_Ejercicio2;
2
3 public class Main {
4
5     public static void main(String[] args) throws Exception {
6
7
8         int numero;
9         try{
10             numero = 'a';
11             System.out.println("Todo ok ");
12         } catch (NumberFormatException ex){
13             System.out.println("Excepcion: " + ex.getMessage());
14         }
15     }
16 }
```

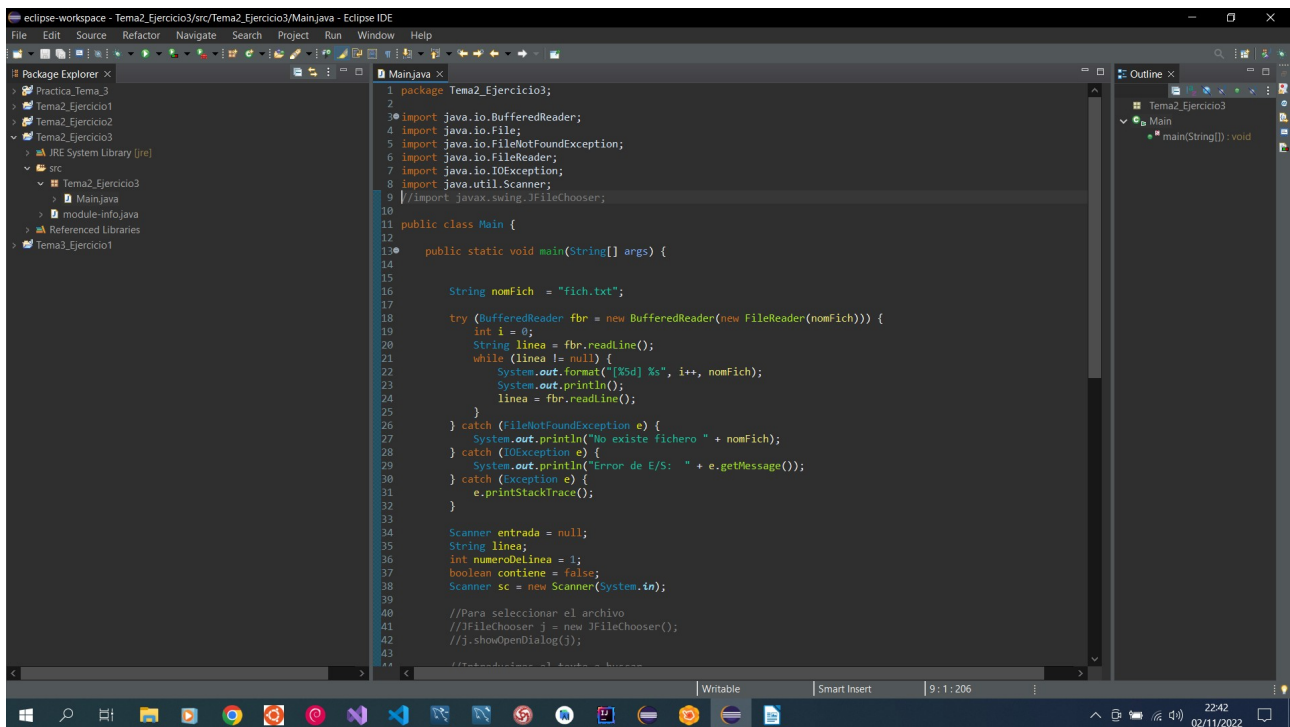
c) Modifica el programa para que haya o no excepción, se imprima siempre el texto “En Acceso a Datos, controlamos nuestro errores”.



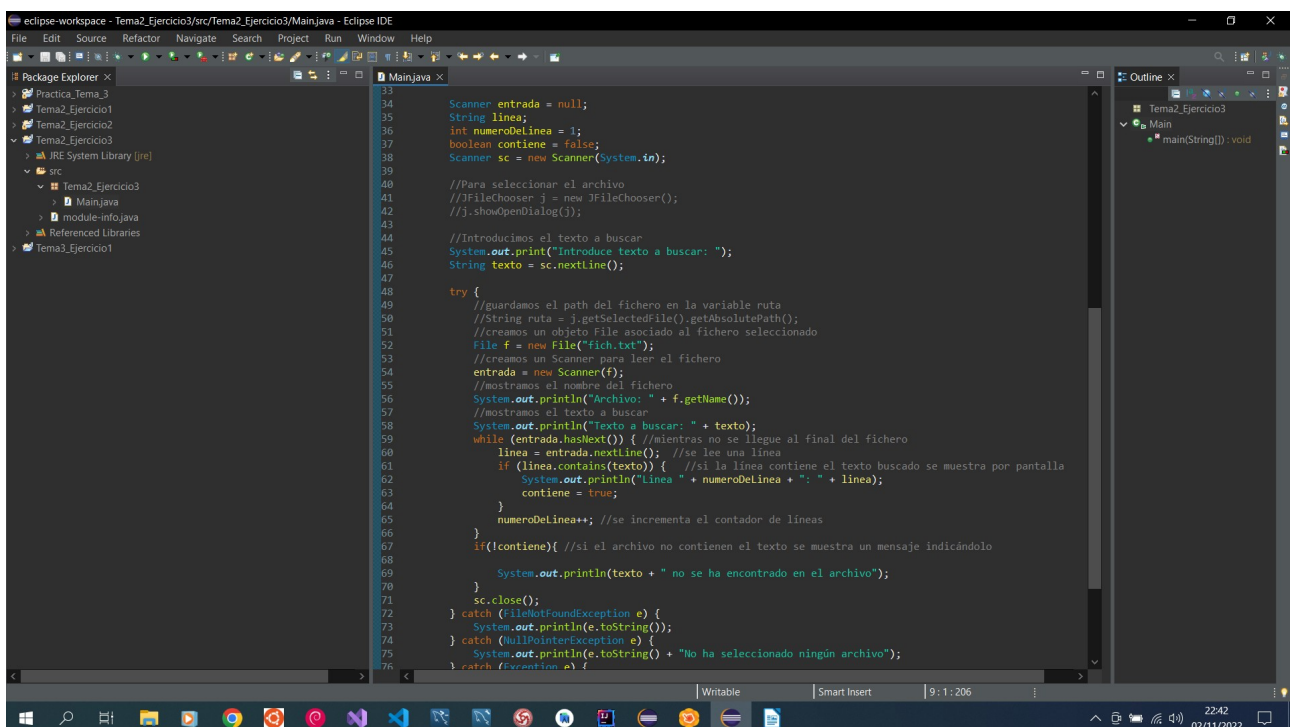
```
1 package Tema2_Ejercicio2;
2
3 public class Main {
4
5     public static void main(String[] args) throws Exception {
6
7
8         int numero;
9         try{
10             numero = 'a';
11             System.out.println("Todo ok ");
12         } catch (NumberFormatException ex){
13             System.out.println("Excepcion: " + ex.getMessage());
14         } finally {
15             System.out.println("En Acceso a Datos, controlamos nuestro errores");
16         }
17     }
18 }
```

## 2.7 – ACCESO SECUENCIAL EN JAVA

a) Crea un programa que busque un texto dado en un fichero de texto, y que muestre para cada aparición la línea y la columna. Se recomienda leer el fichero línea a línea y, dentro de cada línea, buscar las apariciones del texto utilizando un método apropiado de la clase String. Se puede consultar la documentación de dicha clase en la API de Java (<http://docs.oracle.com/javase/8/docs/api>).



```
1 package Tema2_Ejercicio3;
2
3 import java.io.BufferedReader;
4 import java.io.File;
5 import java.io.FileNotFoundException;
6 import java.io.FileReader;
7 import java.io.IOException;
8 import java.util.Scanner;
9 //import javax.swing.JFileChooser;
10
11 public class Main {
12
13     public static void main(String[] args) {
14
15         String nomFich = "fich.txt";
16
17         try (BufferedReader fbr = new BufferedReader(new FileReader(nomFich))) {
18             int i = 0;
19             String linea = fbr.readLine();
20             while (linea != null) {
21                 System.out.format("%5d] %s", i++, nomFich);
22                 System.out.println();
23                 linea = fbr.readLine();
24             }
25         } catch (FileNotFoundException e) {
26             System.out.println("No existe fichero " + nomFich);
27         } catch (IOException e) {
28             System.out.println("Error de E/S: " + e.getMessage());
29         } catch (Exception e) {
30             e.printStackTrace();
31         }
32
33         Scanner entrada = null;
34         String linea;
35         int numeroDeLinea = 1;
36         boolean contiene = false;
37         Scanner sc = new Scanner(System.in);
38
39         //Para seleccionar el archivo
40         //JFileChooser j = new JFileChooser();
41         //j.showOpenDialog();
42
43         //Introducimos el texto a buscar
```



```
33
34 Scanner entrada = null;
35 String linea;
36 int numeroDeLinea = 1;
37 boolean contiene = false;
38 Scanner sc = new Scanner(System.in);
39
40 //Para seleccionar el archivo
41 //JFileChooser j = new JFileChooser();
42 //j.showOpenDialog();
43
44 //Introducimos el texto a buscar
45 System.out.print("Introduce texto a buscar: ");
46 String texto = sc.nextLine();
47
48 try {
49     //guardamos el path del fichero en la variable ruta
50     //String ruta = j.getSelectedFile().getAbsolutePath();
51     //creamos un objeto File asociado al fichero seleccionado
52     File f = new File("fich.txt");
53     //creamos un Scanner para leer el fichero
54     entrada = new Scanner(f);
55     //mostramos el nombre del fichero
56     System.out.println("Archivo: " + f.getName());
57     //mostramos el texto a buscar
58     System.out.println("Texto a buscar: " + texto);
59     while (entrada.hasNext()) { //mientras no se llegue al final del fichero
60         linea = entrada.nextLine(); //se lee una línea
61         if (linea.contains(texto)) { //si la línea contiene el texto buscado se muestra por pantalla
62             System.out.println("Línea " + numeroDeLinea + ": " + linea);
63             contiene = true;
64             numeroDeLinea++; //se incrementa el contador de líneas
65         }
66     }
67     if (!contiene) { //si el archivo no contienen el texto se muestra un mensaje indicándolo
68         System.out.println(texto + " no se ha encontrado en el archivo");
69     }
70     sc.close();
71 } catch (FileNotFoundException e) {
72     System.out.println(e.toString());
73 } catch (NullPointerException e) {
74     System.out.println(e.toString() + " No ha seleccionado ningún archivo");
75 } catch (Exception e) {
76     e.printStackTrace();
77 }
```

```

42 //j.showOpenDialog();
43
44 //Introducimos el texto a buscar
45 System.out.print("Introduce texto a buscar: ");
46 String texto = sc.nextLine();
47
48 try {
49     //guardamos el path del fichero en la variable ruta
50     //String ruta = j.getSelectedFile().getAbsolutePath();
51     //creamos un objeto File asociado al fichero seleccionado
52     File f = new File("fich.txt");
53     //creamos un Scanner para leer el fichero
54     entrada = new Scanner(f);
55     //mostramos el nombre del fichero
56     System.out.println("Archivo: " + f.getName());
57     //mostramos el texto a buscar
58     System.out.println("Texto a buscar: " + texto);
59     while (entrada.hasNext()) { //mientras no se llegue al final del fichero
60         linea = entrada.nextLine(); //se lee una linea
61         if (linea.contains(texto)) { //si la linea contiene el texto buscado se muestra por pantalla
62             System.out.println("Linea " + numeroDeLinea + ": " + linea);
63             contiene = true;
64         }
65         numeroDeLinea++; //se incrementa el contador de lineas
66     }
67     if (!contiene) { //si el archivo no contienen el texto se muestra un mensaje indicándolo
68         System.out.println(texto + " no se ha encontrado en el archivo");
69     }
70     sc.close();
71 } catch (FileNotFoundException e) {
72     System.out.println(e.toString());
73 } catch (NullPointerException e) {
74     System.out.println(e.toString());
75 } catch (Exception e) {
76     System.out.println(e.toString());
77 } finally {
78     if (entrada != null) {
79         entrada.close();
80     }
81 }
82 }
83
84
85

```

## 2.8 – ACCESO ALEATORIO EN JAVA

a) Crear un programa Java que realice la inserción de datos de empleados en un fichero aleatorio llamado “AleatorioEmple.dat”. El programa insertará 7 registros donde para cada empleado se insertarán los siguientes registros:

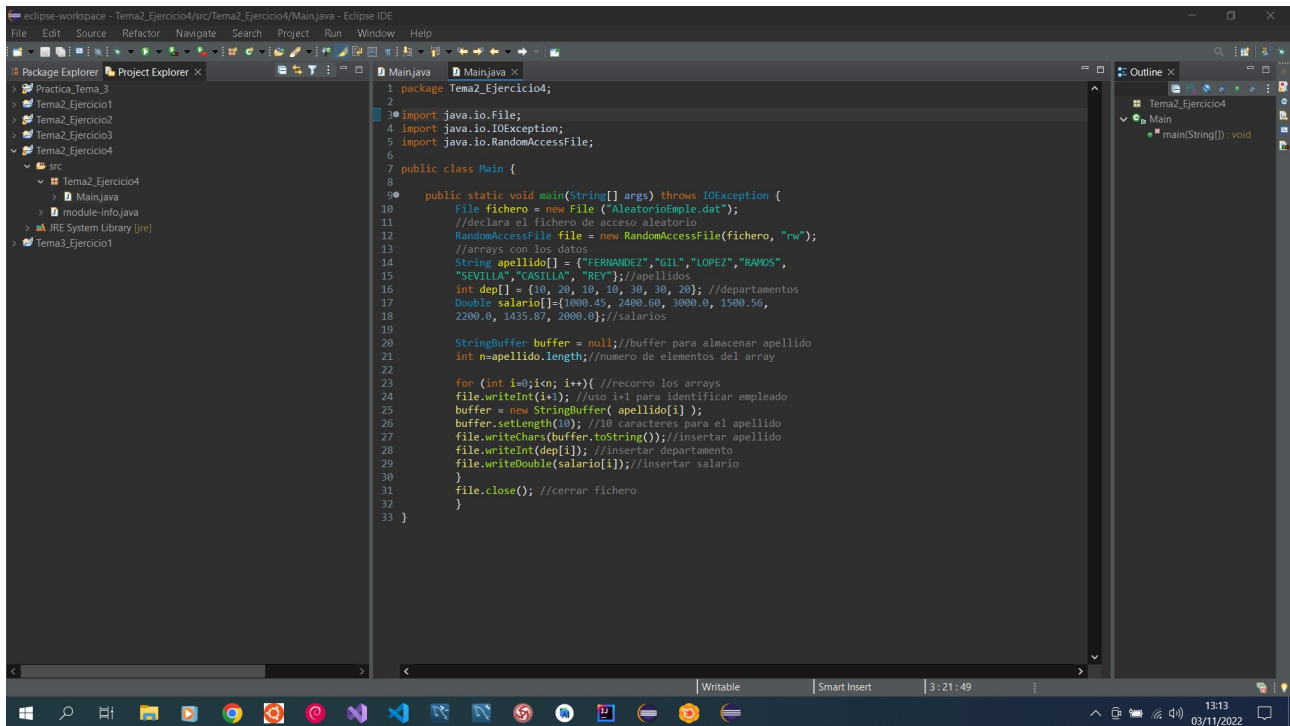
```

1 package Tema2_Ejercicio4;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.io.RandomAccessFile;
6
7 public class Main {
8
9     public static void main(String[] args) throws IOException {
10         File fichero = new File("AleatorioEmple.dat");
11         //declara el fichero de acceso aleatorio
12         RandomAccessFile file = new RandomAccessFile(fichero, "rw");
13         //arrays con los datos
14         String apellido[] = {"FERNANDEZ", "GIL", "LOPEZ", "RAMOS",
15                             "SEVILLA", "CASTILLA", "REY"}; //apellidos
16         int dep[] = {10, 20, 10, 10, 30, 30, 20}; //departamentos
17         Double salario[] = {1000.45, 2400.60, 3000.0, 1500.56,
18                             2200.0, 1435.87, 2000.0}; //salarios
19
20         StringBuffer buffer = null; //buffer para almacenar apellido
21         int n=apellido.length; //numero de elementos del array
22
23         for (int i=0; i<n; i++) { //recorro los arrays
24             file.writeInt(i+1); //uso i+1 para identificar empleado
25             buffer = new StringBuffer( apellido[i] );
26             buffer.setLength(10); //10 caracteres para el apellido
27             file.writeChars(buffer.toString()); //insertar apellido
28             file.writeInt(dep[i]); //insertar departamento
29             file.writeDouble(salario[i]); //insertar salario
30         }
31         file.close(); //cerrar fichero
32     }
33 }

```

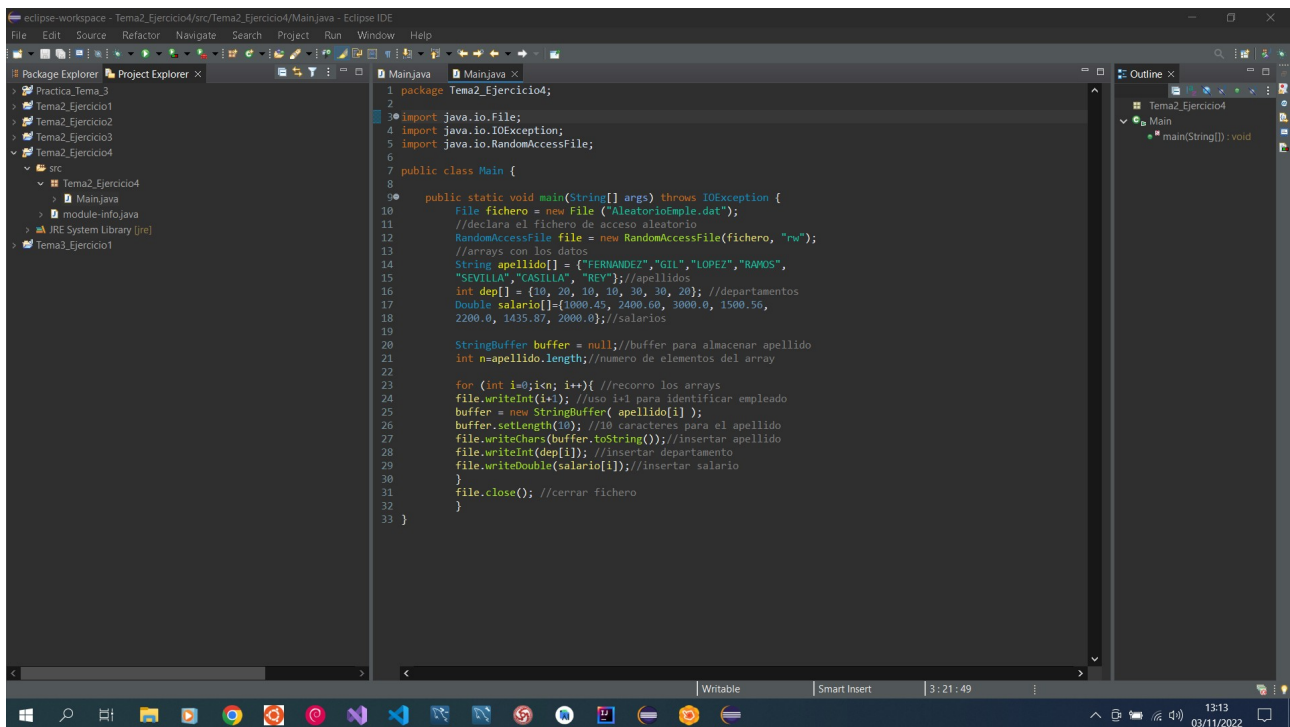


I. Un identificador (Int) que coincidirá con el índice +1 con el que se recorren los arrays.  
Apellido del empleado (String de 10 Caracteres).



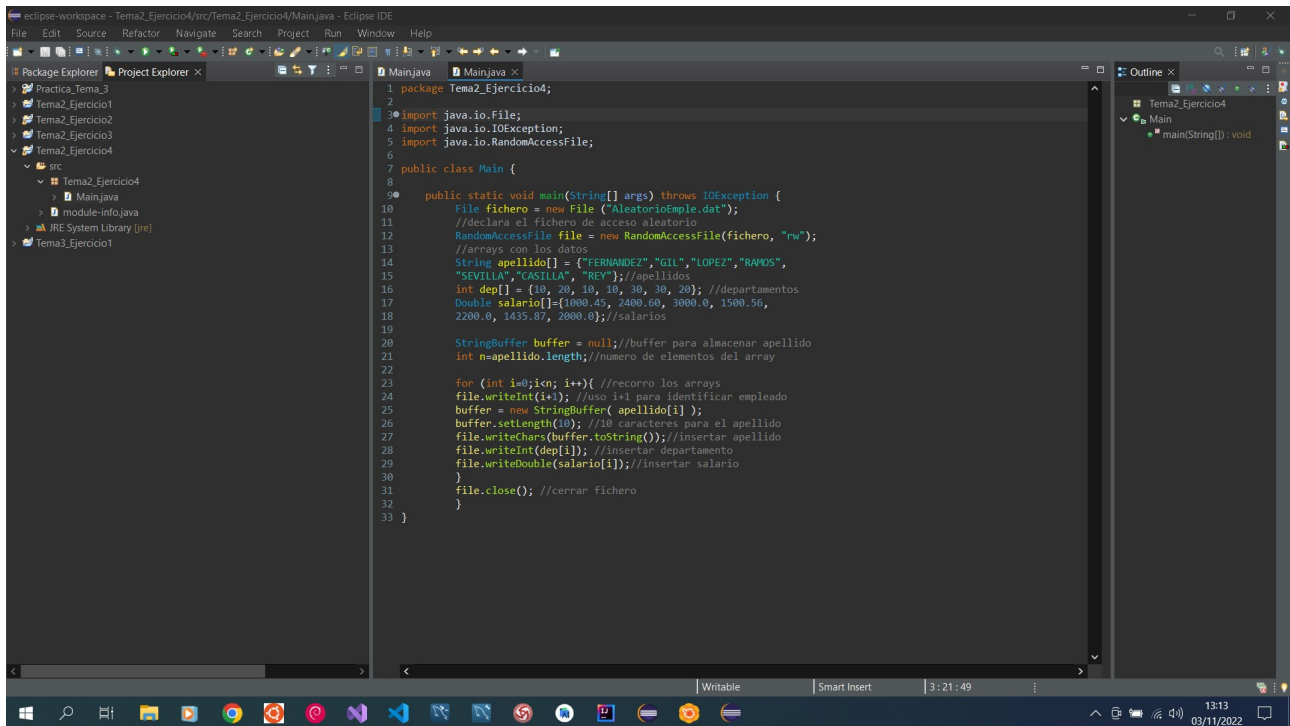
```
1 package Tema2_Ejercicio4;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.io.RandomAccessFile;
6
7 public class Main {
8
9     public static void main(String[] args) throws IOException {
10         File fichero = new File ("AleatorioEmple.dat");
11         //declara el fichero de acceso aleatorio
12         RandomAccessFile file = new RandomAccessFile(fichero, "rw");
13         //arrays con los datos
14         String apellido[] = {"FERNANDEZ","GIL","LOPEZ","RAMOS",
15                             "SEVILLA","CASILLA","REY"}; //apellidos
16         int dep[] = {10, 20, 10, 10, 30, 30, 20}; //departamentos
17         Double salario[] = {1000.45, 2400.60, 3000.0, 1500.56,
18                             2200.0, 1435.87, 2000.0}; //salarios
19
20         StringBuffer buffer = null; //buffer para almacenar apellido
21         int n=apellido.length; //numero de elementos del array
22
23         for (int i=0; i<n; i++){ //recorro los arrays
24             file.writeInt(i+1); //uso i+1 para identificar empleado
25             buffer = new StringBuffer( apellido[i] );
26             buffer.setLength(10); //10 caracteres para el apellido
27             file.writeChars(buffer.toString()); //insertar apellido
28             file.writeInt(dep[i]); //insertar departamento
29             file.writeDouble(salario[i]); //insertar salario
30         }
31         file.close(); //cerrar fichero
32     }
33 }
```

II. Ejemplo: {"FERNANDEZ","GIL ","LOPEZ","RAMOS","SEVILLA","CASILLA",  
"REY"}



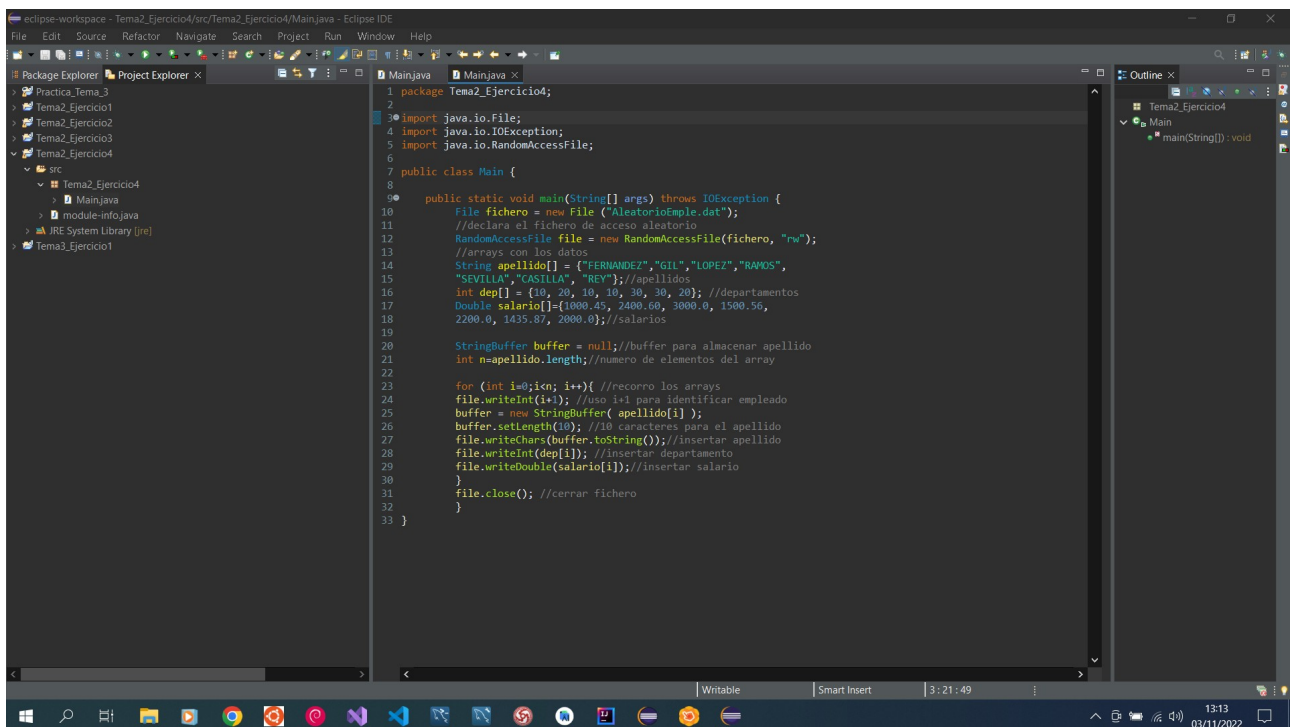
```
1 package Tema2_Ejercicio4;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.io.RandomAccessFile;
6
7 public class Main {
8
9     public static void main(String[] args) throws IOException {
10         File fichero = new File ("AleatorioEmple.dat");
11         //declara el fichero de acceso aleatorio
12         RandomAccessFile file = new RandomAccessFile(fichero, "rw");
13         //arrays con los datos
14         String apellido[] = {"FERNANDEZ","GIL ","LOPEZ","RAMOS",
15                             "SEVILLA","CASILLA","REY"}; //apellidos
16         int dep[] = {10, 20, 10, 10, 30, 30, 20}; //departamentos
17         Double salario[] = {1000.45, 2400.60, 3000.0, 1500.56,
18                             2200.0, 1435.87, 2000.0}; //salarios
19
20         StringBuffer buffer = null; //buffer para almacenar apellido
21         int n=apellido.length; //numero de elementos del array
22
23         for (int i=0; i<n; i++){ //recorro los arrays
24             file.writeInt(i+1); //uso i+1 para identificar empleado
25             buffer = new StringBuffer( apellido[i] );
26             buffer.setLength(10); //10 caracteres para el apellido
27             file.writeChars(buffer.toString()); //insertar apellido
28             file.writeInt(dep[i]); //insertar departamento
29             file.writeDouble(salario[i]); //insertar salario
30         }
31         file.close(); //cerrar fichero
32     }
33 }
```

### III. Código de departamento (Int). Ejemplo: {10, 20, 10, 10, 30, 30, 20}



```
1 package Tema2_Ejercicio4;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.io.RandomAccessFile;
6
7 public class Main {
8
9     public static void main(String[] args) throws IOException {
10         File fichero = new File ("AleatorioEmple.dat");
11         //declara el fichero de acceso aleatorio
12         RandomAccessFile file = new RandomAccessFile(fichero, "rw");
13         //arrays con los datos
14         String apellido[] = {"FERNANDEZ","GIL","LOPEZ","RAMOS",
15                             "SEVILLA","CASILLA", "REY"}; //apellidos
16         int dep[] = {10, 20, 10, 10, 30, 30, 20}; //departamentos
17         Double salario[] = {1000.45, 2400.60, 3000.0, 1500.56,
18                             2200.0, 1435.87, 2000.0}; //salarios
19
20         StringBuffer buffer = null; //buffer para almacenar apellido
21         int n=apellido.length; //numero de elementos del array
22
23         for (int i=0; i<n; i++){ //recorro los arrays
24             file.writeInt(i+1); //uso i+1 para identificar empleado
25             buffer = new StringBuffer( apellido[i] );
26             buffer.setLength(10); //10 caracteres para el apellido
27             file.writeChars(buffer.toString()); //insertar apellido
28             file.writeInt(dep[i]); //insertar departamento
29             file.writeDouble(salario[i]); //insertar salario
30         }
31         file.close(); //cerrar fichero
32     }
33 }
```

### IV. Salario (Double). Ejemplo:{1000.45, 2400.60, 3000.0, 1500.56, 2200.0, 1435.87, 2000.0}



```
1 package Tema2_Ejercicio4;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.io.RandomAccessFile;
6
7 public class Main {
8
9     public static void main(String[] args) throws IOException {
10         File fichero = new File ("AleatorioEmple.dat");
11         //declara el fichero de acceso aleatorio
12         RandomAccessFile file = new RandomAccessFile(fichero, "rw");
13         //arrays con los datos
14         String apellido[] = {"FERNANDEZ","GIL","LOPEZ","RAMOS",
15                             "SEVILLA","CASILLA", "REY"}; //apellidos
16         int dep[] = {10, 20, 10, 10, 30, 30, 20}; //departamentos
17         Double salario[] = {1000.45, 2400.60, 3000.0, 1500.56,
18                             2200.0, 1435.87, 2000.0}; //salarios
19
20         StringBuffer buffer = null; //buffer para almacenar apellido
21         int n=apellido.length; //numero de elementos del array
22
23         for (int i=0; i<n; i++){ //recorro los arrays
24             file.writeInt(i+1); //uso i+1 para identificar empleado
25             buffer = new StringBuffer( apellido[i] );
26             buffer.setLength(10); //10 caracteres para el apellido
27             file.writeChars(buffer.toString()); //insertar apellido
28             file.writeInt(dep[i]); //insertar departamento
29             file.writeDouble(salario[i]); //insertar salario
30         }
31         file.close(); //cerrar fichero
32     }
33 }
```

**V. NOTA: El apellido debe tener/ocupar 10 caracteres. Utilizar la clase StringBuffer para almacenar el apellido.**

```

1 package Tema2_Ejercicio4;
2
3 import java.io.File;
4 import java.io.IOException;
5 import java.io.RandomAccessFile;
6
7 public class Main {
8
9     public static void main(String[] args) throws IOException {
10         File fichero = new File ("Aleatorio.dat");
11         //declara el fichero de acceso aleatorio
12         RandomAccessFile file = new RandomAccessFile(fichero, "rw");
13         //arrays con los datos
14         String apellido[] = {"FERNANDEZ","GIL","LOPEZ","RAMOS",
15                             "SEVILLA","CASILLA","REY"}; //apellidos
16         int dep[] = {10, 20, 10, 10, 30, 20}; //departamentos
17         Double salario[] = {1000.45, 2400.60, 3000.0, 1500.56,
18                             2200.0, 1435.87, 2000.0}; //salarios
19
20         StringBuffer buffer = null; //buffer para almacenar apellido
21         int n=apellido.length; //numero de elementos del array
22
23         for (int i=0; i<n; i++){ //recorro los arrays
24             file.writeInt(i+1); //uso i+1 para identificar empleado
25             buffer = new StringBuffer( apellido[i] );
26             buffer.setLength(10); //10 caracteres para el apellido
27             file.writeChars(buffer.toString()); //insertar apellido
28             file.writeInt(dep[i]); //insertar departamento
29             file.writeDouble(salario[i]); //insertar salario
30         }
31         file.close(); //cerrar fichero
32     }
33 }

```

**b) Modificar el programa anterior para que lea y muestre los datos del fichero de la siguiente forma: i. ID: 1 Apellido: FERNÁNDEZ, Departamento: 10, Salario: 1000.45**

```

29         file.writeChars(salario[i]); //insertar departamento
30         file.writeDouble(salario[i]); //insertar salario
31     }
32     file.close(); //cerrar fichero
33
34     LeerFicheros();
35 }
36
37 public static void LeerFicheros() throws IOException {
38     File fichero = new File ("Aleatorio.dat");
39     //declara el fichero de acceso aleatorio
40     RandomAccessFile file = new RandomAccessFile(fichero, "r");
41     //
42     int id, dep, posicion;
43     Double salario;
44     char apellido[] = new char[10], aux;
45     posicion = 0; //para situarnos al principio
46     while (file.getFilePointer() != file.length()) {
47         file.seek(posicion); //nos posicionamos en posicion
48         id = file.readInt(); //obtengo id de empleado
49
50         //recorro uno a uno los caracteres del apellido
51         for (int i = 0; i < apellido.length; i++) {
52             aux = file.readChar();
53             apellido[i] = aux; //los voy guardando en el array
54         }
55         //convierto a String el array
56         String apellidos = new String(apellido);
57         dep = file.readInt(); //obtengo dep
58         salario = file.readDouble(); //obtengo salario
59
60         if(id > 0)
61             System.out.printf("ID: %s, Apellido: %s, Departamento: %d, Salario: %.2f\n",
62                               id, apellidos.trim(), dep, salario);
63
64         //me posiciono para el sig empleado, cada empleado ocupa 36 bytes
65         posicion= posicion + 36;
66     } //fin bucle while
67
68     file.close(); //cerrar fichero
69 }
70
71 }

```

Console Output:

```

-terminated- Main (7) [Java Application] C:\Users\david\workspace\plataforma\ejercicios\Tema2_Ejercicio4\Main.java
ID: 1, Apellido: FERNANDEZ, Departamento: 10, Salario: 1000.45
ID: 2, Apellido: GIL, Departamento: 20, Salario: 2400.60
ID: 3, Apellido: LOPEZ, Departamento: 10, Salario: 3000.00
ID: 4, Apellido: RAMOS, Departamento: 10, Salario: 1500.56
ID: 5, Apellido: SEVILLA, Departamento: 30, Salario: 2200.00
ID: 6, Apellido: CASILLA, Departamento: 30, Salario: 1435.87
ID: 7, Apellido: REY, Departamento: 20, Salario: 2000.00

```