

Entornos de desarrollo

Bloque 2

Tema 9: Acceso a ficheros de texto

Soluciones

Ejercicios propuestos:

2.9.1.1. Crea un programa que pida dos frases al usuario y las guarde en un fichero llamado "dosFrases.txt".

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

public class DosFrases {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        System.out.print("Introduce una frase: ");
        String frase1 = teclado.nextLine();

        System.out.print("Introduce otra frase: ");
        String frase2 = teclado.nextLine();

        PrintWriter printWriter = null;
        try{
            printWriter = new PrintWriter("dosFrases.txt");
            printWriter.println(frase1);
            printWriter.println(frase2);
            printWriter.close();
        }catch (IOException e){
            e.printStackTrace();
        }finally{
            if(printWriter != null){
                printWriter.close();
            }
        }
        teclado.close();
    }
}
```

2.9.1.2. Crea un programa que pida frases al usuario (una cantidad indeterminada, hasta que introduzca una línea vacía) y las almacene en un fichero llamado "frases.txt". Cada vez que se lance el programa, se destruirá el fichero "frases.txt" y se creará uno nuevo que lo reemplace.

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

public class Frases {

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

        PrintWriter printWriter = null;
        try{
            printWriter = new PrintWriter("frases.txt");
            do{
                System.out.print("Introduce una frase: ");
                frase = teclado.nextLine();
                if(!frase.isEmpty()){
                    printWriter.println(frase);
                }
            }while(!frase.isEmpty());
            printWriter.close();
        }catch (IOException e){
            e.printStackTrace();
        }finally{
            if(printWriter != null){
                printWriter.close();
            }
        }
        teclado.close();
    }
}
```

2.9.1.3. Crea una variante del ejercicio anterior, en la que el fichero se llamará "anotaciones.txt" y no se destruirá en cada nueva ejecución, sino que se añadirán las nuevas frases al final de las existentes.

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

public class Anotaciones {

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

        PrintWriter printWriter = null;
        try{
            // Abrimos el fichero en modo adición
            printWriter = new PrintWriter(
                new BufferedWriter(
                    new FileWriter("anotaciones.txt", true)
                )
            );
            do{
                System.out.print("Introduce una frase: ");
                frase = teclado.nextLine();
                if(!frase.isEmpty()){
                    printWriter.println(frase);
                }
            }while(!frase.isEmpty());
            printWriter.close();
        }catch (IOException e){
            e.printStackTrace();
        }finally{
            if(printWriter != null){
                printWriter.close();
            }
        }
        teclado.close();
    }
}
```

2.9.1.4. Crea una versión mejorada del ejercicio anterior, en la que antes de cada frase se anotará la fecha y la hora en la que se ha realizado dicha anotación. Ayúdate de `LocalDateTime.now()` para conseguir la fecha.

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;
import java.io.PrintWriter;
import java.time.LocalDateTime;
import java.util.Scanner;

public class Anotaciones2 {

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

        PrintWriter printWriter = null;
        try{
            // Abrimos el fichero en modo adición
            printWriter = new PrintWriter(
                new BufferedWriter(
                    new FileWriter("frases.txt", true)
                )
            );
            do{
                System.out.print("Introduce una frase: ");
                frase = teclado.nextLine();
                if(!frase.isEmpty()){
                    printWriter.println(
                        LocalDateTime.now() + " - " + frase);
                }
            }while(!frase.isEmpty());
            printWriter.close();
        }catch (IOException e){
            e.printStackTrace();
        }finally{
            if(printWriter != null){
                printWriter.close();
            }
        }
        teclado.close();
    }
}
```

2.9.1.5. Crea un programa que pida al usuario una anchura y una altura y cree un fichero llamado "rectangulo.txt" que contenga un rectángulo de asteriscos de esa anchura y el altura. Por ejemplo, si el usuario indica anchura 5 y altura 3, el fichero contendría:

```
*****
*****
*****
```

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

public class Rectangulo {

    @SuppressWarnings("resource")
    public static void main(String[] args) {
        int ancho = 0;
        int alto = 0;

        Scanner entrada = new Scanner(System.in);
        System.out.print("Introduce la anchura del rectángulo: ");
        while(!entrada.hasNextInt()){
            System.out.print("Un número entero por favor: ");
            entrada.nextLine();
        }
        ancho = entrada.nextInt();

        entrada = new Scanner(System.in);
        System.out.print("Introduce la altura del rectángulo: ");
        while(!entrada.hasNextInt()){
            System.out.print("Un número entero por favor: ");
            entrada.nextLine();
        }
        alto = entrada.nextInt();
        entrada.close();

        PrintWriter printWriter = null;
        try{
            printWriter = new PrintWriter("rectangulo.txt");
            for(int f = 0; f < alto; f++){
                for(int c = 0; c < ancho; c++){
                    printWriter.print("*");
                }
                printWriter.println();
            }
            printWriter.close();
        }catch (IOException e){
            e.printStackTrace();
        }finally{
```

```
        if(printWriter != null){  
            printWriter.close();  
        }  
    }  
}
```

2.9.1.6. Crea un programa que pregunte al usuario la cantidad de días de un mes, el número del primer día (1 para lunes, 7 para domingo) y el nombre del mes (por ejemplo, "Marzo") y cree un fichero de texto llamado "agendaMarzo.txt" (o el nombre del mes que corresponde), que contendrá un esqueleto de agenda para ese mes. Por ejemplo, si el mes es Septiembre, tiene 31 días y empieza en el día 4, el contenido del fichero sería algo como:

Septiembre

Jueves 1:

Viernes 2:

Sábado 3:

Domingo 4:

Lunes 5:

(...)

Viernes 30:

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

public class Agenda {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        String mes = null;
        int num_dias = 0;
        int dia = 0; // 1 para lunes, 7 para domingo

        System.out.print("Introduce un mes: ");
        mes = teclado.next();
        System.out.print("Introduce el número de días: ");
        num_dias = teclado.nextInt();
        System.out.print("Introduce el número del día de inicio (1-7): ");
        dia = teclado.nextInt();

        PrintWriter printWriter = null;
        try{
            printWriter = new PrintWriter("agenda"+mes+".txt");
            printWriter.println(mes);
            printWriter.println();
        }
```

```
        printWriter.println("-----");
        -----");

        for(int i = 1; i <= num_dias; i++){
            switch(dia){
                case 1:
                    printWriter.print("Lunes ");
                    break;
                case 2:
                    printWriter.print("Martes ");
                    break;
                case 3:
                    printWriter.print("Miércoles ");
                    break;
                case 4:
                    printWriter.print("Jueves ");
                    break;
                case 5:
                    printWriter.print("Viernes ");
                    break;
                case 6:
                    printWriter.print("Sábado ");
                    break;
                case 7:
                    printWriter.print("Domingo ");
                    break;
            }
            printWriter.println(i+":");
            printWriter.println("-----");
            -----");
            dia++;
            if(dia>7)
                dia=1;
        }

        printWriter.close();
    }catch (IOException e){
        e.printStackTrace();
    }finally{
        if(printWriter != null){
            printWriter.close();
        }
    }
    teclado.close();
}
}
```


2.9.1.7. Crea un programa que pregunte al usuario la cantidad de días de un mes, el número del primer día (1 para lunes, 7 para domingo) y el nombre del mes (por ejemplo, "Marzo") y cree un fichero de texto llamado "calendarioMarzo.txt" (o el nombre del mes que corresponde), con el calendario de ese mes. Por ejemplo, si el mes es Septiembre, tiene 31 días y empieza en el día 4, el contenido del fichero sería algo como:

Septiembre

lun mar mie jue vie sab dom

			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

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

public class Calendario {

    public static void main(String[] args) {
        Scanner teclado = new Scanner(System.in);
        String mes = null;
        int num_dias = 0;
        int dia = 0; // 1 para lunes, 7 para domingo

        System.out.print("Introduce un mes: ");
        mes = teclado.next();
        System.out.print("Introduce el número de días: ");
        num_dias = teclado.nextInt();
        System.out.print("Introduce el número del día de inicio (1-7): ");
        dia = teclado.nextInt();

        PrintWriter printWriter = null;
        try{
            printWriter = new PrintWriter("calendario"+mes+".txt");
            printWriter.println(mes);
            printWriter.println();
            printWriter.println("lun mar mie jue vie sab dom");
            printWriter.println();

            if(dia > 1){
                for(int i = 1; i < dia; i++){
                    printWriter.print("    ");
                }
            }
        }
```

```
        for(int i = 1; i <= num_dias; i++){
            if(i < 10)
                printWriter.print(" "+i+" ");
            else printWriter.print(" "+i+" ");
            dia++;
            if(dia>7){
                dia=1;
                printWriter.println();
            }
        }

        printWriter.close();
    }catch (IOException e){
        e.printStackTrace();
    }finally{
        if(printWriter != null){
            printWriter.close();
        }
    }
    teclado.close();
    System.out.println("Fin.");
}
}
```

2.9.2.1. Crea un programa que muestre la primera línea del contenido del fichero "DosFrases.txt".

```
import java.io.*;

public class DosFrases {

    public static void main(String[] args) {
        // Primero vemos si el fichero existe
        if(!(new File("dosFrases.txt")).exists()){
            System.out.println("No he encontrado dosFrases.txt");
        }else{
            // En caso de que exista, intentamos leer
            System.out.println("Leyendo fichero dosFrases.txt...");
            System.out.println("-----");
            try{
                BufferedReader ficheroEntrada = new BufferedReader(
                    new FileReader(new File("dosFrases.txt")))
                );

                String linea = ficheroEntrada.readLine();

                if(linea != null)
                    System.out.println(linea);
                else System.out.println("El fichero está vacío.");

                ficheroEntrada.close();
            }catch(IOException errorDeFichero){
                System.out.println("Ha habido problemas: "
                    + errorDeFichero.getMessage());
            }
            System.out.println("-----");
            System.out.println("Fin de la lectura.");
        }
    }
}
```

2.9.2.2. Crea un programa que muestre todo el contenido del fichero "anotaciones.txt".

```
import java.io.*;

public class Anotaciones {

    public static void main(String[] args) {
        // Primero vemos si el fichero existe
        if(!(new File("anotaciones.txt")).exists()){
            System.out.println("No he encontrado anotaciones.txt");
        }else{
            // En caso de que exista, intentamos leer
            System.out.println("Leyendo fichero anotaciones.txt...");
            System.out.println("-----");
            try{
                BufferedReader ficheroEntrada = new BufferedReader(
                    new FileReader(new File("anotaciones.txt"))
                );

                String linea = ficheroEntrada.readLine();

                if(linea == null)
                    System.out.println("El fichero está vacío.");

                while(linea != null){
                    System.out.println(linea);
                    linea = ficheroEntrada.readLine();
                }

                ficheroEntrada.close();
            }catch(IOException errorDeFichero){
                System.out.println("Ha habido problemas: "
                    + errorDeFichero.getMessage());
            }
            System.out.println("-----");
            System.out.println("Fin de la lectura.");
        }
    }
}
```

2.9.2.3. Crea un programa que muestre paginado el contenido del fichero "anotaciones.txt": tras cada 23 líneas se realizará una pausa hasta que el usuario pulse Intro.

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

public class AnotacionesPaginado {

    public static void main(String[] args) {
        // Primero vemos si el fichero existe
        if(!(new File("anotaciones.txt")).exists()){
            System.out.println("No he encontrado anotaciones.txt");
        }else{
            // En caso de que exista, intentamos leer
            System.out.println("Leyendo fichero anotaciones.txt...");
            System.out.println("-----");
            try{
                BufferedReader ficheroEntrada = new BufferedReader(
                    new FileReader(new File("anotaciones.txt"))
                );

                String linea = ficheroEntrada.readLine();

                if(linea == null)
                    System.out.println("El fichero está vacío.");

                Scanner teclado = new Scanner(System.in);
                int contador = 1;
                while(linea != null){
                    System.out.println(contador+" "+linea);
                    contador++;
                    if(contador > 23){
                        contador = 1;
                        System.out.print("Pulsa INTRO para continuar...");
                        @SuppressWarnings("unused")
                        String tecla = teclado.nextLine();
                    }
                    linea = ficheroEntrada.readLine();
                }
                teclado.close();

                ficheroEntrada.close();
            }catch(IOException errorDeFichero){
                System.out.println("Ha habido problemas: "
                    + errorDeFichero.getMessage());
            }
            System.out.println("-----");
            System.out.println("Fin de la lectura.");
        }
    }
}
```

```
}  
}
```

2.9.2.4. Crea un programa que muestre el contenido de un fichero de texto, cuyo nombre deberá introducir el usuario. Debe avisar si el fichero no existe.

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class MuestraFichero {

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

        System.out.print("Nombre del fichero (extensión incluida): ");
        String fichero = entrada.nextLine();

        // Primero vemos si el fichero existe
        if(!(new File(fichero)).exists()){
            System.out.println("No he encontrado el fichero " + fichero);
        }else{
            // En caso de que exista, intentamos leer
            System.out.println("Leyendo fichero anotaciones.txt...");
            System.out.println("-----");

            try{
                BufferedReader ficheroEntrada = new BufferedReader(
                    new FileReader(new File(fichero))
                );

                String linea = ficheroEntrada.readLine();

                if(linea == null)
                    System.out.println("El fichero está vacío.");

                while(linea != null){
                    System.out.println(linea);
                    linea = ficheroEntrada.readLine();
                }

                ficheroEntrada.close();
            }catch(IOException errorDeFichero){
                System.out.println("Ha habido problemas: "
                    + errorDeFichero.getMessage());
            }

            System.out.println("-----");
            System.out.println("Fin de la lectura.");
        }
    }
}
```

```
        System.out.println("\nFin del programa.\n");  
        entrada.close();  
    }  
}
```


2.9.2.5. Crea un programa que lea el contenido de un fichero de texto y lo vuelque a otro fichero de texto, pero convirtiendo cada línea a mayúsculas.

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.io.PrintWriter;
import java.util.Scanner;

public class VolcadoFichero {

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

        System.out.print("Fichero a leer (extensión incluida): ");
        String fichero = entrada.nextLine();

        // Primero vemos si el fichero existe
        if(!(new File(fichero)).exists()){
            System.out.println("No he encontrado el fichero " + fichero);
        }else{
            // En caso de que exista, intentamos leer
            System.out.println("Leyendo fichero anotaciones.txt...");

            try{
                BufferedReader ficheroEntrada = new BufferedReader(
                    new FileReader(new File(fichero))
                );
                PrintWriter ficheroSalida = null;
                ficheroSalida= new PrintWriter("copia" + fichero);

                String linea = ficheroEntrada.readLine();

                if(linea == null)
                    System.out.println("El fichero está vacío.");

                while(linea != null){
                    ficheroSalida.println(linea.toUpperCase());
                    linea = ficheroEntrada.readLine();
                }

                ficheroEntrada.close();
                ficheroSalida.close();
            }catch(IOException errorDeFichero){
                System.out.println("Ha habido problemas: "
                    + errorDeFichero.getMessage());
            }
        }
    }
}
```

```
        System.out.println("Fin de la lectura.");  
    }  
    System.out.println("\nFin del programa.\n");  
    entrada.close();  
}  
}
```

2.9.2.6. Crea un programa que pida al usuario el nombre de un fichero y una palabra a buscar en él. Debe mostrar en pantalla todas las líneas del fichero que contengan esa palabra.

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;

public class BuscaEnFichero {

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

        System.out.print("Fichero a leer (extensión incluida): ");
        String fichero = entrada.nextLine();

        // Primero vemos si el fichero existe
        if(!(new File(fichero)).exists()){
            System.out.println("No he encontrado el fichero " + fichero);
        }else{
            System.out.print("Palabra a buscar en el fichero: ");
            String palabra = entrada.next();
            System.out.println("Palabra a buscar: " + palabra);

            System.out.println("\nBuscando...\n");

            try{
                BufferedReader ficheroEntrada = new BufferedReader(
                    new FileReader(new File(fichero))
                );

                String linea = ficheroEntrada.readLine();

                if(linea == null){
                    System.out.println("El fichero está vacío.");
                }else{
                    int cont = 1;
                    while(linea != null){
                        if(linea.indexOf(palabra) != -1){
                            System.out.println("Línea "+cont+": "+linea);
                        }
                        linea = ficheroEntrada.readLine();
                        cont++;
                    }
                }

                ficheroEntrada.close();
            }catch(IOException errorDeFichero){
```

```
        System.out.println("Ha habido problemas: "
            + errorDeFichero.getMessage());
    }
}
System.out.println("\nFin del programa.\n");
entrada.close();
}
}
```

2.9.2.7. Crea un programa que lea el contenido del fichero "rectangulo.txt" que creaste en el ejercicio 2.9.1.5. y que calcule y muestre en pantalla cual es la anchura y la altura del rectángulo de asteriscos.

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;

public class Rectangulo2 {

    public static void main(String[] args) {
        // Primero vemos si el fichero existe
        if(!(new File("rectangulo.txt")).exists()){
            System.out.println("No se ha encontrado rectangulo.txt");
        }else{
            int ancho = 0;
            int alto = 0;

            BufferedReader ficheroEntrada = null;
            try{
                System.out.println("Leyendo el fichero rectangulo.txt");
                ficheroEntrada = new BufferedReader(
                    new FileReader(new File("rectangulo.txt")));

                String linea = ficheroEntrada.readLine();

                while(linea != null){
                    alto++;
                    if(linea.length() > ancho)
                        ancho=linea.length();

                    linea = ficheroEntrada.readLine();
                }

                ficheroEntrada.close();
            }catch (IOException e){
                e.printStackTrace();
            }

            System.out.println("Ancho: " + ancho);
            System.out.print("Alto.: " + alto);
        }
    }
}
```

2.9.2.8. Crea un programa que pida al usuario el nombre de un fichero. Almacenará el contenido del fichero en un ArrayList y, a partir de ese momento, de forma repetitiva, pedirá al usuario una palabra a buscar y mostrará en pantalla todas las líneas contengan la contengan, o el texto "No encontrada" según corresponda. Se repetirá hasta que el usuario introduzca una cadena vacía.

```
import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Scanner;

public class Buscador {

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

        System.out.print("Fichero a leer (extensión incluida): ");
        String fichero = entrada.nextLine();

        // Primero vemos si el fichero existe
        if(!(new File(fichero)).exists()){
            System.out.println("No he encontrado el fichero " + fichero);
        }else{
            ArrayList<String> lineas = new ArrayList<String>();
            try{
                BufferedReader ficheroEntrada = new BufferedReader(
                    new FileReader(new File(fichero))
                );

                String linea = ficheroEntrada.readLine();

                if(linea == null){
                    System.out.println("El fichero está vacío.");
                }else{
                    while(linea != null){
                        lineas.add(linea);
                        linea = ficheroEntrada.readLine();
                    }
                }
                ficheroEntrada.close();
            }catch(IOException errorDeFichero){
                System.out.println("Ha habido problemas: "
                    + errorDeFichero.getMessage());
            }

            String palabra;
            do {
                System.out.print("\nPalabra a buscar en el fichero: ");
```

```
        palabra = entrada.nextLine();

        if(!palabra.isEmpty()) {
            System.out.println("Palabra a buscar: " + palabra);
            System.out.println("\nBuscando...\n");
            boolean existe = false;
            for(int i = 0; i < lineas.size(); i++) {
                if(lineas.get(i).contains(palabra)) {
                    System.out.println(lineas.get(i));
                    existe = true;
                }
            }
            if(!existe)
                System.out.println("No encontrada");
        }
    }while(!palabra.isEmpty());
}
System.out.println("\nFin del programa.\n");
entrada.close();
}
```