

Práctica 1 - Expresiones Regulares en Java

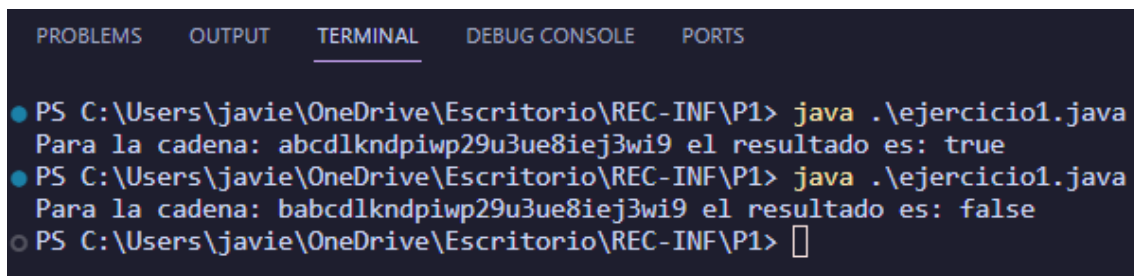
Ejercicio 1. Comprobar si una cadena empieza por “abc”.

Expresión regular usada: “^abc.*”



```
J ejercicio1.java X
J ejercicio1.java > ejercicio1
1 import java.util.regex.Matcher;
2 import java.util.regex.Pattern;
3
4 public class ejercicio1 {
5     public static void main(String[] args) {
6         String cadena = "babcdlkndpiwp29u3ue8iej3wi9";
7
8         Pattern pat = Pattern.compile(regex:"^abc.*");
9         Matcher mat = pat.matcher(cadena);
10
11         System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
12     }
13 }
```

Figura 1 Código del ejercicio 1



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio1.java
Para la cadena: abcdlkndpiwp29u3ue8iej3wi9 el resultado es: true
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio1.java
Para la cadena: babcdlkndpiwp29u3ue8iej3wi9 el resultado es: false
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> 
```

Figura 2 Ejemplo de ejecución del ejercicio 1

Como podemos ver en la salida, cuando la cadena empieza por abc, nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

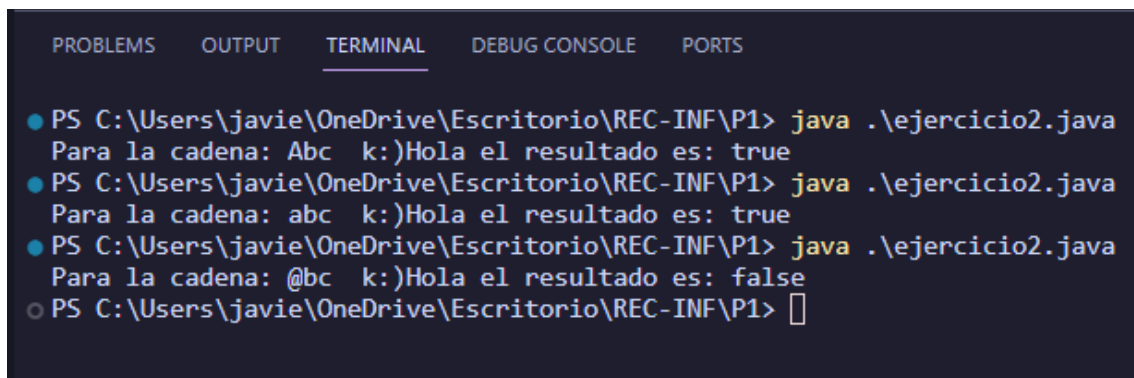
Ejercicio 2. Comprobar si una cadena empieza por “abc” o “Abc”.

Expresión regular usada: “^(a|A)bc.*”



```
J ejercicio2.java X
J ejercicio2.java > ejercicio2
1 import java.util.regex.Matcher;
2 import java.util.regex.Pattern;
3
4 public class ejercicio2 {
5     public static void main(String[] args) {
6         String cadena = "Abc k:)Hola";
7
8         Pattern pat = Pattern.compile(regex:"^(a|A)bc.*");
9         Matcher mat = pat.matcher(cadena);
10
11         System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
12     }
13 }
14
```

Figura 3 Código del ejercicio 2



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio2.java
Para la cadena: Abc k:)Hola el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio2.java
Para la cadena: abc k:)Hola el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio2.java
Para la cadena: @bc k:)Hola el resultado es: false
○ PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1>
```

Figura 4 Ejemplo de ejecución del ejercicio 2

Como podemos ver en la salida, cuando la cadena empieza por Abc o abc, nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

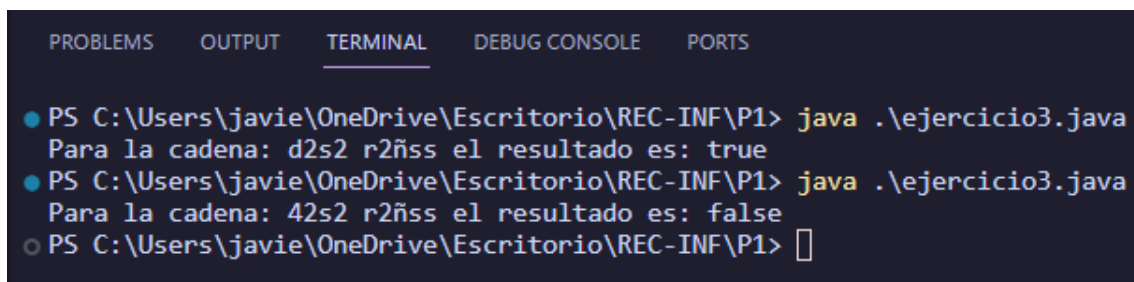
Ejercicio 3. Comprobar si una cadena no empieza por un dígito.

Expresión regular usada: “`^\d.*`”



```
J ejercicio3.java X
J ejercicio3.java > ejercicio3 > main(String[])
1 import java.util.regex.Matcher;
2 import java.util.regex.Pattern;
3
4 public class ejercicio3 {
5     public static void main(String[] args) {
6         String cadena = "2sss";
7
8         Pattern pat = Pattern.compile(regex: "^\d.*");
9         Matcher mat = pat.matcher(cadena);
10
11         System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
12     }
13 }
14
```

Figura 5 Código del ejercicio 3



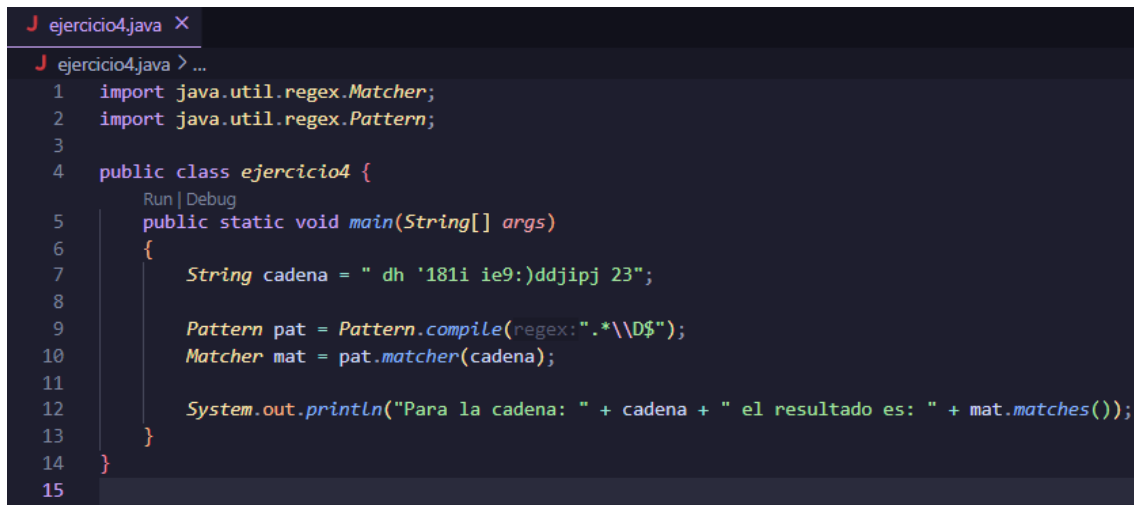
```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio3.java
Para la cadena: d2s2 r2ñss el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio3.java
Para la cadena: 42s2 r2ñss el resultado es: false
○ PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> 
```

Figura 6 Ejemplo de ejecución del ejercicio 3

Como podemos ver en la salida, cuando la cadena no empieza por un número, nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

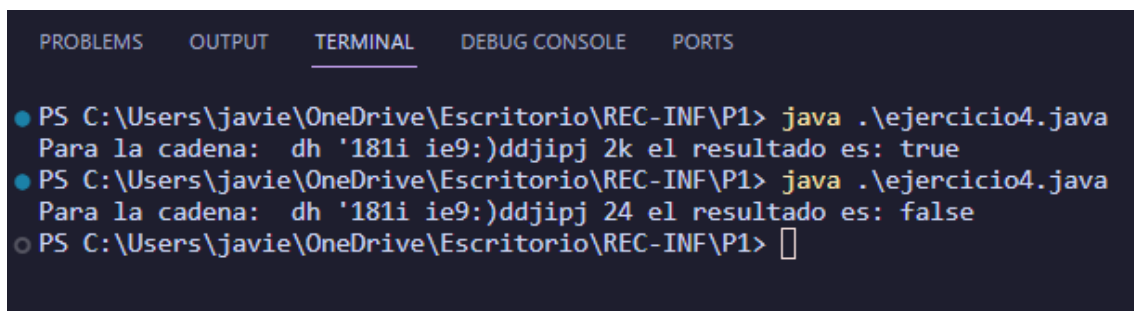
Ejercicio 4. Comprobar si una cadena no acaba con un dígito.

Expresión regular usada: “.*\\D\$”



```
ejercicio4.java X
ejercicio4.java > ...
1  import java.util.regex.Matcher;
2  import java.util.regex.Pattern;
3
4  public class ejercicio4 {
5      Run | Debug
6      public static void main(String[] args)
7      {
8          String cadena = " dh '181i ie9:)ddjipj 23";
9
10         Pattern pat = Pattern.compile(regex:".*\\D$");
11         Matcher mat = pat.matcher(cadena);
12
13         System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
14     }
15 }
```

Figura 7 Código del ejercicio 4



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio4.java
Para la cadena: dh '181i ie9:)ddjipj 2k el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio4.java
Para la cadena: dh '181i ie9:)ddjipj 24 el resultado es: false
○ PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> 
```

Figura 8 Ejemplo de ejecución del ejercicio 4

Como podemos ver en la salida, cuando la cadena no termina por un número, nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

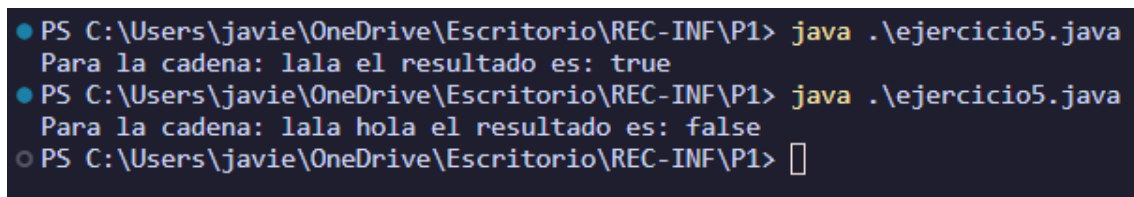
Ejercicio 5. Comprobar si una cadena solo contiene los caracteres “l” o “a”.

Expresión regular usada: “^(l|a)+\$”



```
ejercicio5.java X
ejercicio5.java > ...
1  import java.util.regex.Matcher;
2  import java.util.regex.Pattern;
3
4  public class ejercicio5 {
5      public static void main(String[] args)
6      {
7          String cadena = "lallllllllla";
8
9          Pattern pat = Pattern.compile(regex: "^(l|a)+$");
10         Matcher mat = pat.matcher(cadena);
11
12         System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
13     }
14 }
15
```

Figura 9 Código del ejercicio 5



```
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio5.java
Para la cadena: lala el resultado es: true
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio5.java
Para la cadena: lala hola el resultado es: false
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1>
```

Figura 10 Ejemplo de ejecución del ejercicio 5

Como podemos ver en la salida, cuando la cadena solo tiene caracteres “a” o “l”, nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

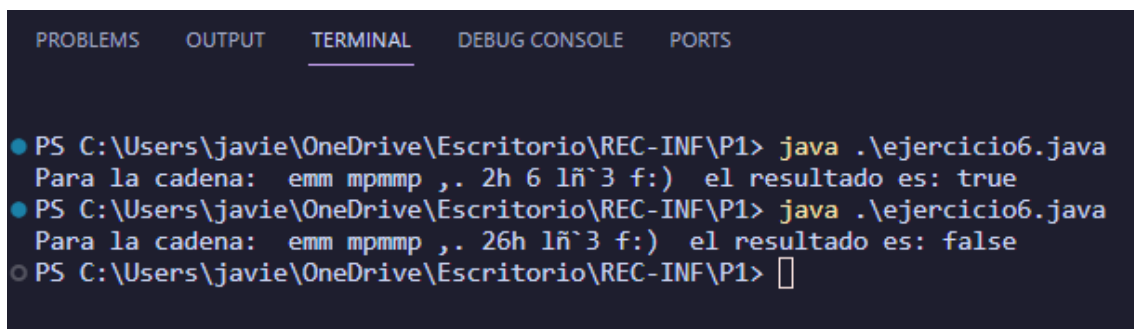
Ejercicio 6. Comprobar si una cadena contiene un 2 y ese no está seguido por un 6.

Expresión regular usada: `".*2[^6].*"`



```
ejercicio6.java X
ejercicio6.java > ...
1  import java.util.regex.Matcher;
2  import java.util.regex.Pattern;
3
4  public class ejercicio6 {
5      public static void main(String[] args)
6      {
7          String cadena = " emm mpmp ,. 26 lñ`3 f:) ";
8
9          Pattern pat = Pattern.compile(regex:".*2[^6].*");
10         Matcher mat = pat.matcher(cadena);
11
12         System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
13     }
14 }
15
```

Figura 11 Código del ejercicio 6



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio6.java
Para la cadena: emm mpmp ,. 2h 6 lñ`3 f:) el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio6.java
Para la cadena: emm mpmp ,. 26h lñ`3 f:) el resultado es: false
○ PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> 
```

Figura 12 Ejemplo de ejecución del ejercicio 6

Como podemos ver en la salida, cuando la cadena posee un 2 y este no es seguido de un 6, nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

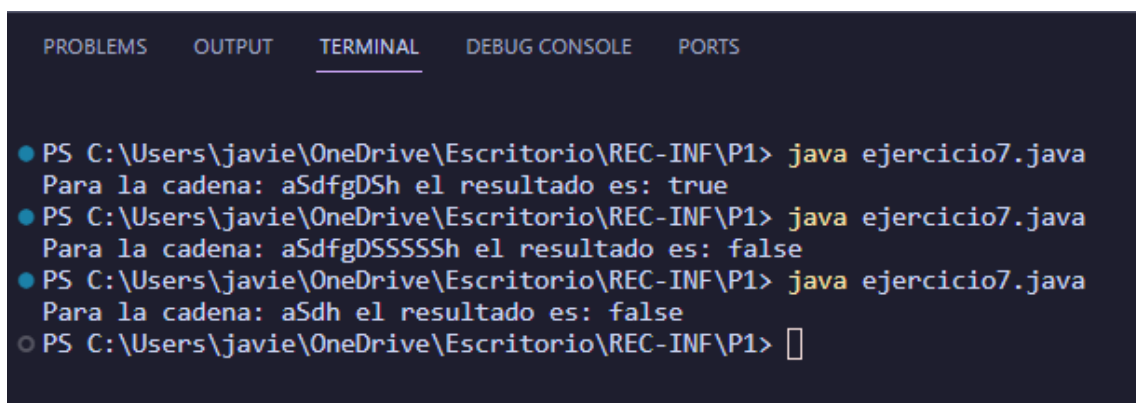
Ejercicio 7. Comprobar si una cadena está formada por un mínimo de 5 letras mayúsculas o minúsculas y un máximo de 10.

Expresión regular usada: `^[a-zA-Z]{5,10}$`



```
ejercicio7.java X
ejercicio7.java > ...
1 import java.util.regex.Matcher;
2 import java.util.regex.Pattern;
3
4 public class ejercicio7 {
5     public static void main(String[] args)
6     {
7         String cadena = "aSdfgDSh";
8
9         Pattern pat = Pattern.compile(regex:"^[a-zA-Z]{5,10}$");
10        Matcher mat = pat.matcher(cadena);
11
12        System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
13    }
14 }
15
```

Figura 13 Código del ejercicio 7



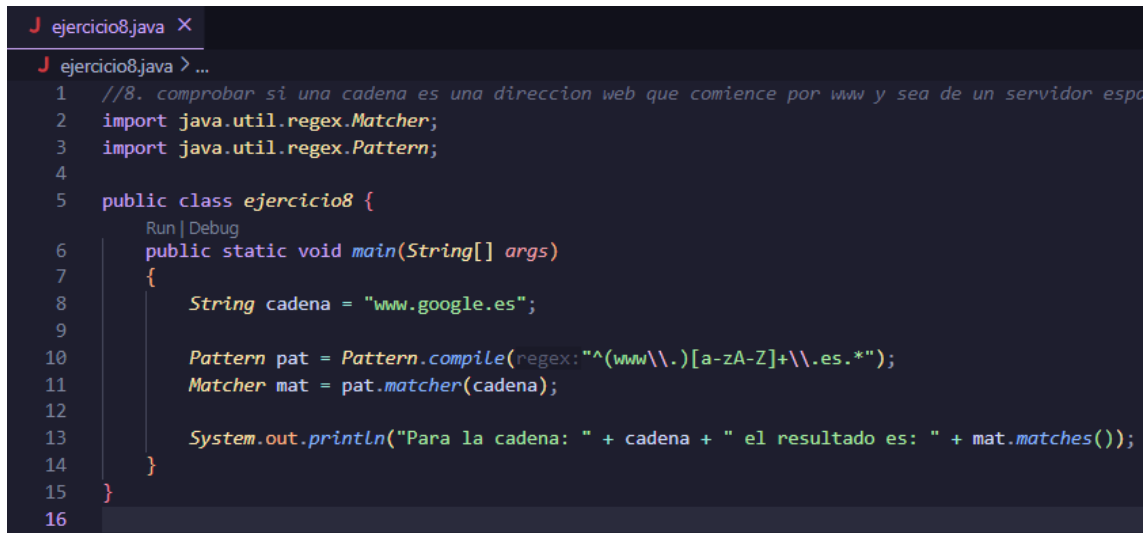
```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java ejercicio7.java
Para la cadena: aSdfgDSh el resultado es: true
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java ejercicio7.java
Para la cadena: aSdfgDSSSSSh el resultado es: false
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java ejercicio7.java
Para la cadena: aSdh el resultado es: false
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1>
```

Figura 14 Ejemplo de ejecución del ejercicio 7

Como podemos ver en la salida, cuando la cadena posee de 5 a 10 palabras (de la “a” la “z” incluyendo mayúsculas), nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

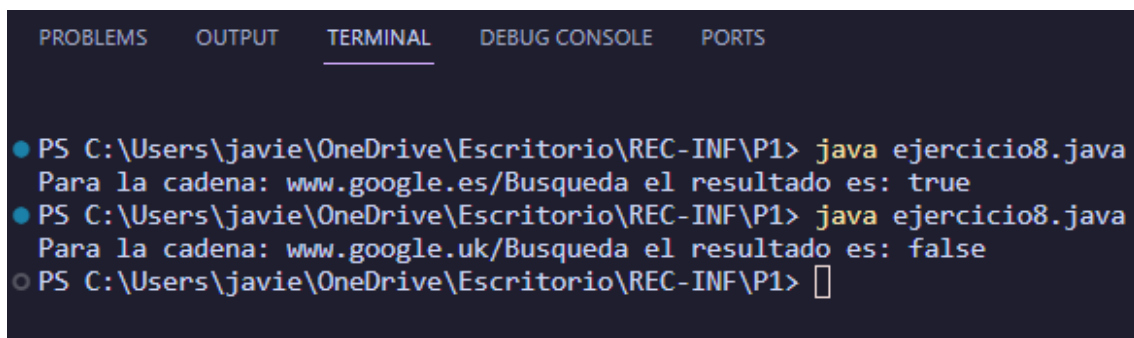
Ejercicio 8. Comprobar si una cadena es una dirección web que comience por www y sea de un servidor español.

Expresión regular usada: “^(www\\.)([a-zA-Z]+\\.es.*)”



```
J ejercicio8.java X
J ejercicio8.java > ...
1 //8. comprobar si una cadena es una direccion web que comience por www y sea de un servidor espa
2 import java.util.regex.Matcher;
3 import java.util.regex.Pattern;
4
5 public class ejercicio8 {
6     public static void main(String[] args)
7     {
8         String cadena = "www.google.es";
9
10        Pattern pat = Pattern.compile(regex: "^(www\\.)([a-zA-Z]+\\.es.*)");
11        Matcher mat = pat.matcher(cadena);
12
13        System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
14    }
15 }
16
```

Figura 15 Código del ejercicio 8



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java ejercicio8.java
Para la cadena: www.google.es/Busqueda el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java ejercicio8.java
Para la cadena: www.google.uk/Busqueda el resultado es: false
○ PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> 
```

Figura 16 Ejemplo de ejecución del ejercicio 8

Como podemos ver en la salida, cuando la cadena posee las tres “www” y la terminación de la web es “.es”, nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

*Nota: He asumido que un servidor español terminaría en es, pero, si incluyese términos como “.com”, “.org” o similares, se podrían modelar con un OR.

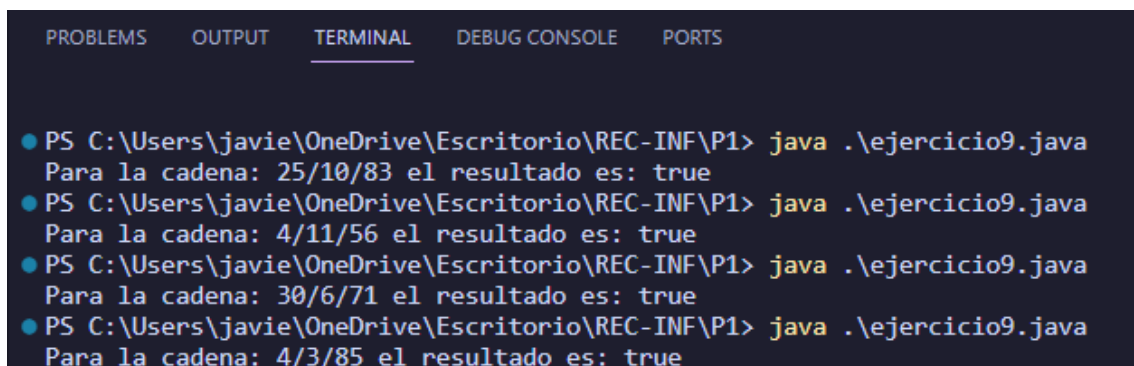
Ejercicio 9. Comprobar si una cadena es una fecha dd/mm/yy. Comprueba que tu patrón coincida con las siguientes fechas: 25/10/83, 4/11/56, 30/6/71 y 4/3/85

Expresión regular usada: “^([1-9]|[1-2][0-9]|3[0-1])/([1-9]|1[0-2])/([1-9]|[0-9][0-9])\$”



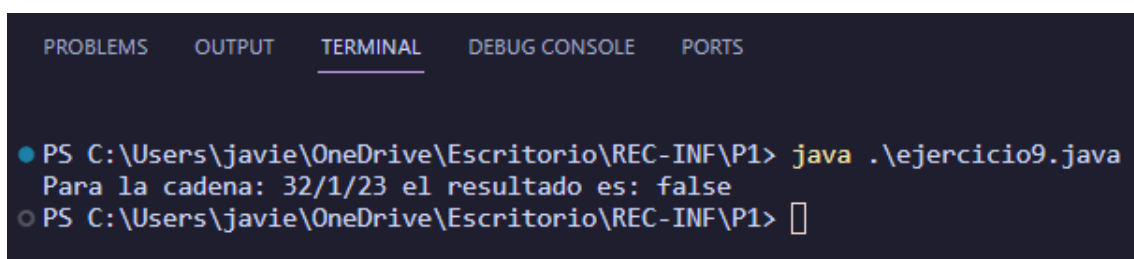
```
J ejercicio9.java X
J ejercicio9.java > ...
1  import java.util.regex.Matcher;
2  import java.util.regex.Pattern;
3
4  //25/10/83, 4/11/56, 30/6/71 y 4/3/85
5  public class ejercicio9 {
6      Run | Debug
7      public static void main(String[] args)
8      {
9          String cadena = "32/1/23";
10
11          Pattern pat = Pattern.compile(regex:"^([1-9]|[1-2][0-9]|3[0-1])/([1-9]|1[0-2])/([1-9]|[0-9][0-9])$");
12          Matcher mat = pat.matcher(cadena);
13
14          System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
15      }
16  }
```

Figura 17 Código del ejercicio 9



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio9.java
Para la cadena: 25/10/83 el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio9.java
Para la cadena: 4/11/56 el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio9.java
Para la cadena: 30/6/71 el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio9.java
Para la cadena: 4/3/85 el resultado es: true
```

Figura 18 Ejemplo de ejecución del ejercicio 8 (I)



```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio9.java
Para la cadena: 32/1/23 el resultado es: false
○ PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> 
```

Figura 19 Ejemplo de ejecución del ejercicio 8 (II)

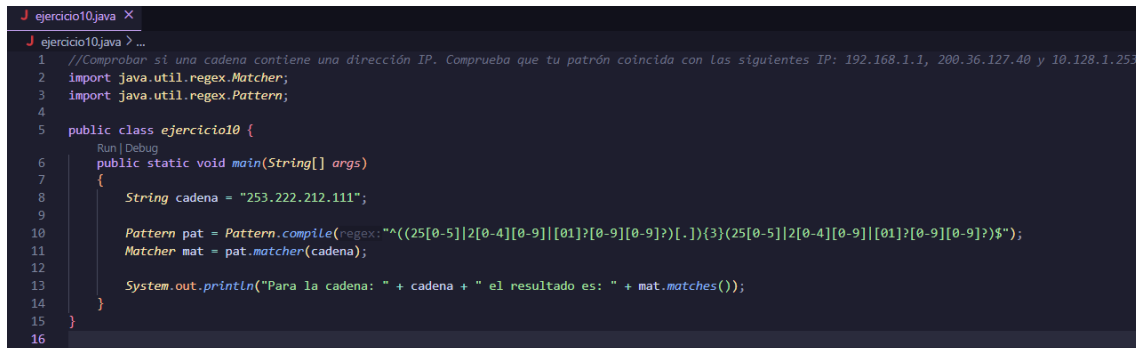
Como podemos ver en la salida, cuando la cadena tiene un formato correcto, nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

*Nota: Se ha querido restringir al máximo las fechas incorrectas, como se puede ver en la Figura 19, una fecha con días > 31 o meses > 12 dará false.

Ejercicio 10. Comprobar si una cadena contiene una dirección IP.

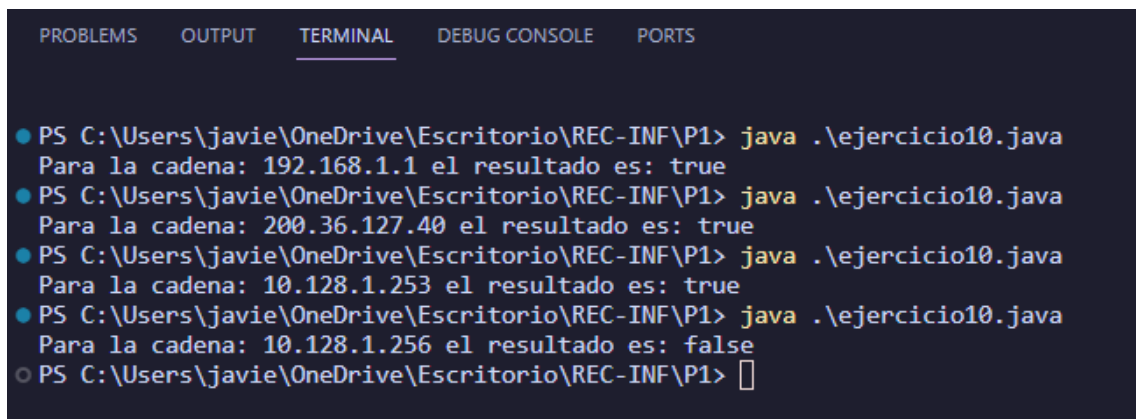
Comprueba que tu patrón coincida con las siguientes IP: 192.168.1.1, 200.36.127.40 y 10.128.1.253.

Expresión regular usada: “^((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)(.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\$”



```
1 //Comprobar si una cadena contiene una dirección IP. Comprueba que tu patrón coincida con las siguientes IP: 192.168.1.1, 200.36.127.40 y 10.128.1.253
2 import java.util.regex.Matcher;
3 import java.util.regex.Pattern;
4
5 public class ejercicio10 {
6     public static void main(String[] args)
7     {
8         String cadena = "253.222.212.111";
9
10        Pattern pat = Pattern.compile(regex: "^((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)(.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)$");
11        Matcher mat = pat.matcher(cadena);
12
13        System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
14    }
15 }
16
```

Figura 20 Código del ejercicio 10



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio10.java
Para la cadena: 192.168.1.1 el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio10.java
Para la cadena: 200.36.127.40 el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio10.java
Para la cadena: 10.128.1.253 el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio10.java
Para la cadena: 10.128.1.256 el resultado es: false
○ PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> 
```

Figura 21 Ejemplo de ejecución del ejercicio 10

Como podemos ver en la salida, cuando la cadena tiene un formato correcto, nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

*Nota: Se ha restringido valores incorrectos en los rangos de la IP, es decir, si uno de sus valores supera el 255, dará error como podemos ver en la Figura 21.

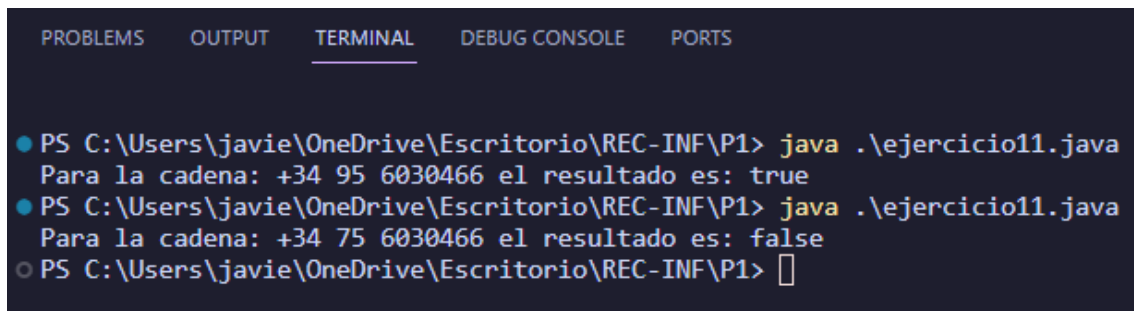
Ejercicio 11. ¿Qué expresión regular utilizarías para comprobar si un número de teléfono fijo es español? Ten en cuenta el siguiente ejemplo para realizar el patrón: +34 95 6030466.

Expresión regular usada: “^\\+34 [89][1-8] \\d{7}\$”



```
ejercicio11.java X
ejercicio11.java > ...
1 import java.util.regex.Matcher;
2 import java.util.regex.Pattern;
3
4 public class ejercicio11 {
5     public static void main(String[] args)
6     {
7         String cadena = "+34 95 6030466";
8
9         Pattern pat = Pattern.compile(regex: "\\+34 [89][1-8] \\d{7}$");
10        Matcher mat = pat.matcher(cadena);
11
12        System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
13    }
14 }
15
```

Figura 22 Código del ejercicio 11



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio11.java
Para la cadena: +34 95 6030466 el resultado es: true
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio11.java
Para la cadena: +34 75 6030466 el resultado es: false
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> 
```

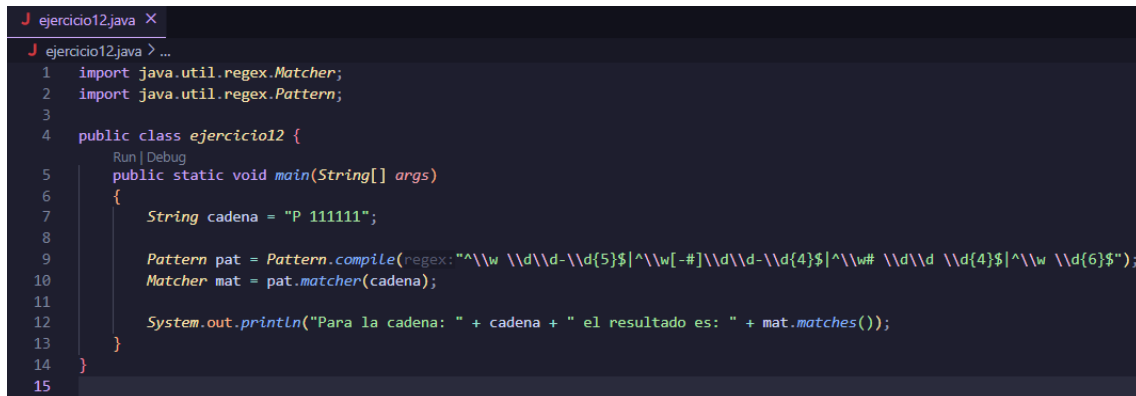
Figura 23 Ejemplo de ejecución del ejercicio 11

Como podemos ver en la salida, cuando la cadena tiene un formato correcto, nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

*Nota: Se ha restringido valores incorrectos, es decir, si el primer número del teléfono no es 8 o 9 y el segundo no es un número entre el 1 y el 8, dará error como podemos ver en la Figura 23.

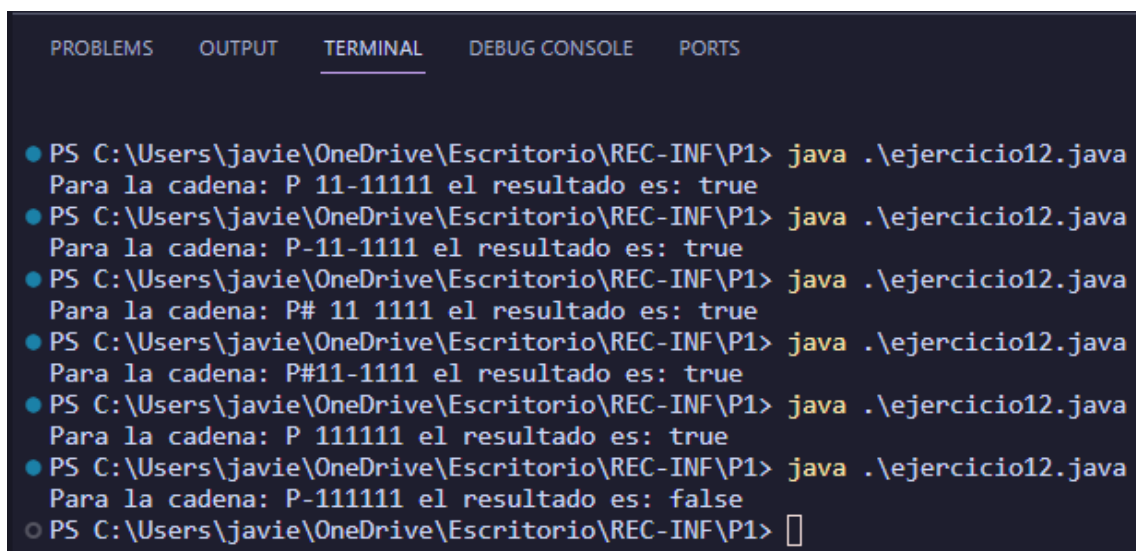
Ejercicio 12. ¿Qué expresión regular utilizarías para comprobar el número de pedido de una empresa? (Para consultar formato consultar documento de prácticas)

Expresión regular usada: “`^\\w \\d\\d-\\d{5}$|^\\w[-#]\\d\\d-\\d{4}$|^\\w# \\d\\d \\d{4}$|^\\w \\d{6}$`”



```
1 import java.util.regex.Matcher;
2 import java.util.regex.Pattern;
3
4 public class ejercicio12 {
5     public static void main(String[] args)
6     {
7         String cadena = "P 111111";
8
9         Pattern pat = Pattern.compile(regex: "^\\w \\d\\d-\\d{5}$|^\\w[-#]\\d\\d-\\d{4}$|^\\w# \\d\\d \\d{4}$|^\\w \\d{6}$");
10        Matcher mat = pat.matcher(cadena);
11
12        System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
13    }
14 }
15
```

Figura 24 Código del ejercicio 12



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio12.java
Para la cadena: P 11-11111 el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio12.java
Para la cadena: P-11-1111 el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio12.java
Para la cadena: P# 11 1111 el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio12.java
Para la cadena: P#11-1111 el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio12.java
Para la cadena: P 111111 el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio12.java
Para la cadena: P-111111 el resultado es: false
○ PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> 
```

Figura 25 Ejemplo de ejecución del ejercicio 12

Como podemos ver en la salida, cuando la cadena tiene un formato correcto, nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

*Nota: Se ha restringido valores incorrectos, es decir, si no se cumple estrictamente alguno de los tipos de formatos de pedidos, dará error como podemos ver en la Figura 25.

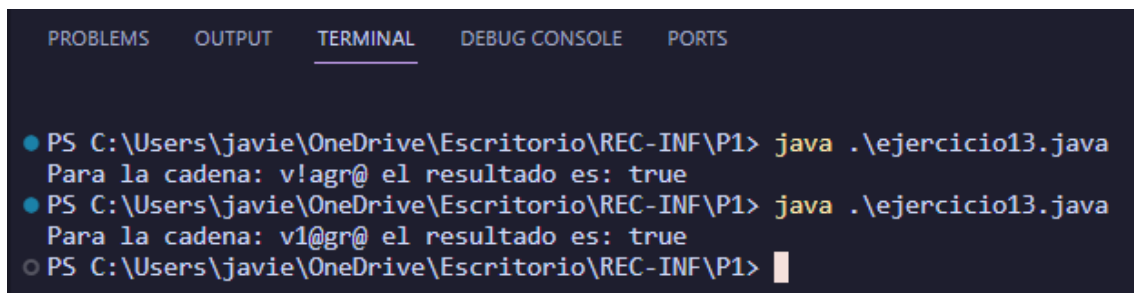
Ejercicio 13. Para evitar el spam, intenta localizar posibles alteraciones que se utilizan para saltarse los filtros de correo. (Para consultar formato consultar documento de prácticas)

Expresión regular usada: “.*v[i!][a@]gr[a@].*”



```
ejercicio13.java X
ejercicio13.java > ...
1  import java.util.regex.Matcher;
2  import java.util.regex.Pattern;
3
4  public class ejercicio13 {
5      Run | Debug
6      public static void main(String[] args)
7      {
8          String cadena = "v!agr@";
9
10         Pattern pat = Pattern.compile(regex: ".*v[i!][a@]gr[a@].*");
11         Matcher mat = pat.matcher(cadena);
12
13         System.out.println("Para la cadena: " + cadena + " el resultado es: " + mat.matches());
14     }
15 }
```

Figura 26 Código del ejercicio 13



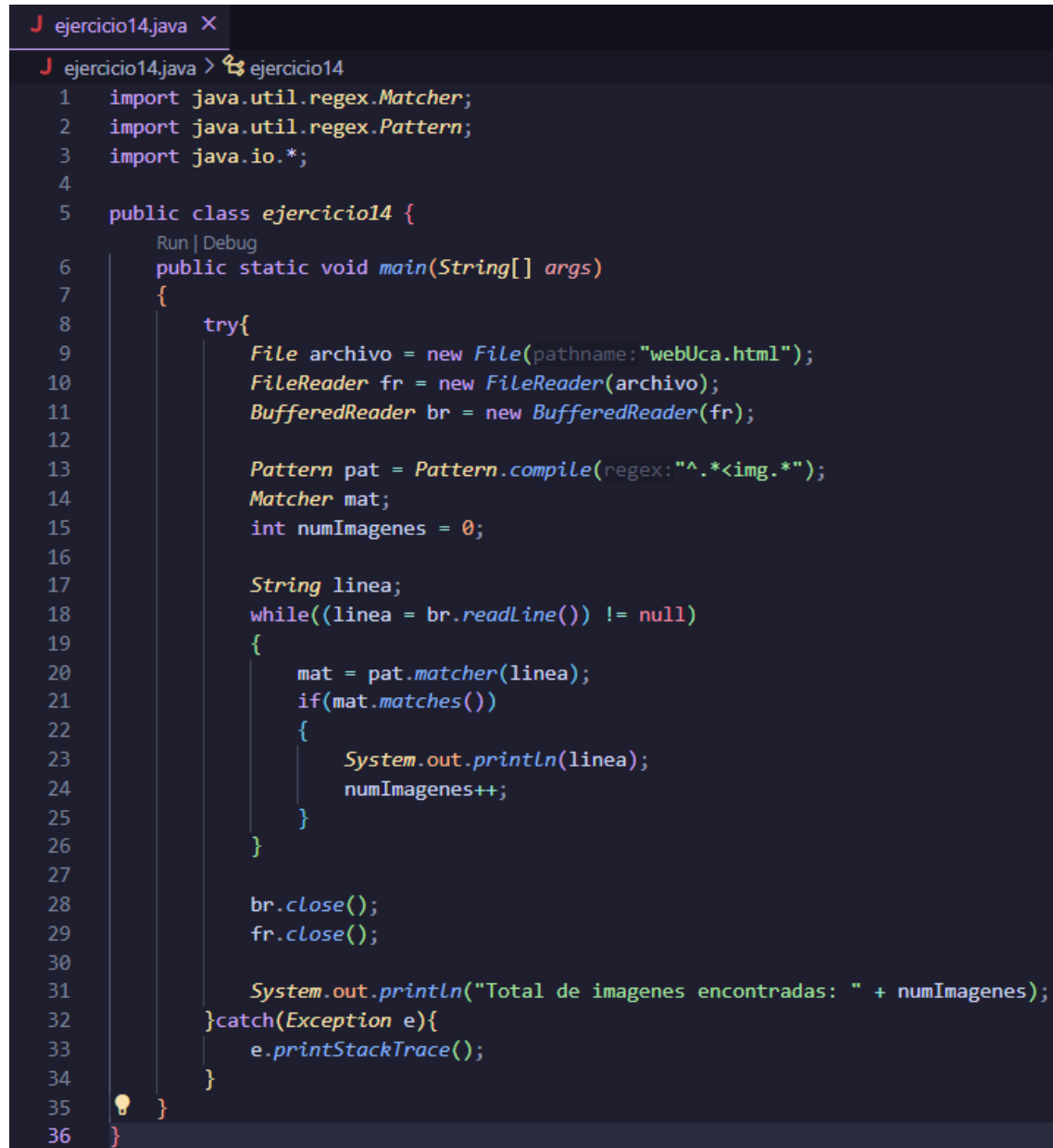
```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio13.java
Para la cadena: v!agr@ el resultado es: true
● PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio13.java
Para la cadena: v1@gr@ el resultado es: true
○ PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> 
```

Figura 27 Ejemplo de ejecución del ejercicio 13

Como podemos ver en la salida, cuando la cadena tiene un formato correcto, nos devuelve un resultado correcto (true) y cuando no, devuelve un resultado incorrecto (false).

Ejercicio 14. Descarga la página principal de la UCA y localiza a través de una expresión regular en el fichero html almacenado, todas las imágenes de la página web. Busca cómo se incluyen imágenes en html.

Expresión regular usada: “^.*<img.*”



```
J ejercicio14.java X
J ejercicio14.java > ejercicio14
1  import java.util.regex.Matcher;
2  import java.util.regex.Pattern;
3  import java.io.*;
4
5  public class ejercicio14 {
6      Run | Debug
7      public static void main(String[] args)
8      {
9          try{
10             File archivo = new File(pathname:"webUca.html");
11             FileReader fr = new FileReader(archivo);
12             BufferedReader br = new BufferedReader(fr);
13
14             Pattern pat = Pattern.compile(regex:"^.*<img.*");
15             Matcher mat;
16             int numImagenes = 0;
17
18             String linea;
19             while((linea = br.readLine()) != null)
20             {
21                 mat = pat.matcher(linea);
22                 if(mat.matches())
23                 {
24                     System.out.println(linea);
25                     numImagenes++;
26                 }
27             }
28
29             br.close();
30             fr.close();
31
32             System.out.println("Total de imagenes encontradas: " + numImagenes);
33         }catch(Exception e){
34             e.printStackTrace();
35         }
36     }
```

Figura 28 Código del ejercicio 14

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE  PORTS














Total de imagenes encontradas: 50
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> |
```

Figura 29 Ejemplo de ejecución del ejercicio 14



Figura 30 Número de imágenes (50 + 1 por error)

Como podemos ver en la salida, nos aparece cada línea del html que contiene una etiqueta de imagen, siendo en total unas 50.

*Nota: Como se puede leer en el título de la Figura 30, existe un error en la estructura del html, donde debería haber un salto de línea, no lo hay, por lo que no reconoce esa imagen.

Ejercicio 15. Consulte el enunciado en el documento de la práctica.

*Nota: se ha intentado resolver usando el método Split propuesto en la propia práctica.

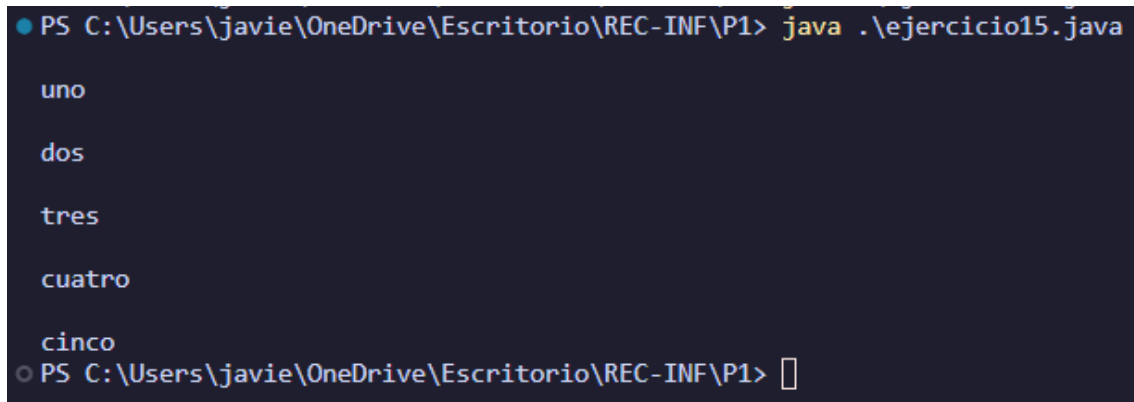


```
1 import java.util.regex.Pattern;
2
3 public class ejercicio15 {
4     public static void main(String[] args) {
5
6         String cadena = "<a>uno</a><b>dos</b><c>tres</c><d>cuatro</d><e>cinco</e>";
7
8         Pattern p = Pattern.compile(regex: "<[^>]*(<[^<]*>)/[>]*>");
9         String[] items = p.split(cadena);
10
11         for(String s : items) {
12             System.out.println(s);
13         }
14     }
15 }
```

Figura 31 Código del ejercicio 15

Usando el patrón: “<[>]*(<[<]*>)/[>]*>”, no obtenemos ningún tipo de salida.

Patrón usado como posible solución: “<\V?\w>”, obtenemos la siguiente salida:



```
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio15.java
uno
dos
tres
cuatro
cinco
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1>
```

Figura 32 Salida obtenida

Usando el patrón “<.*>(.*)<\V.*>”, no obtenemos ningún tipo de salida.

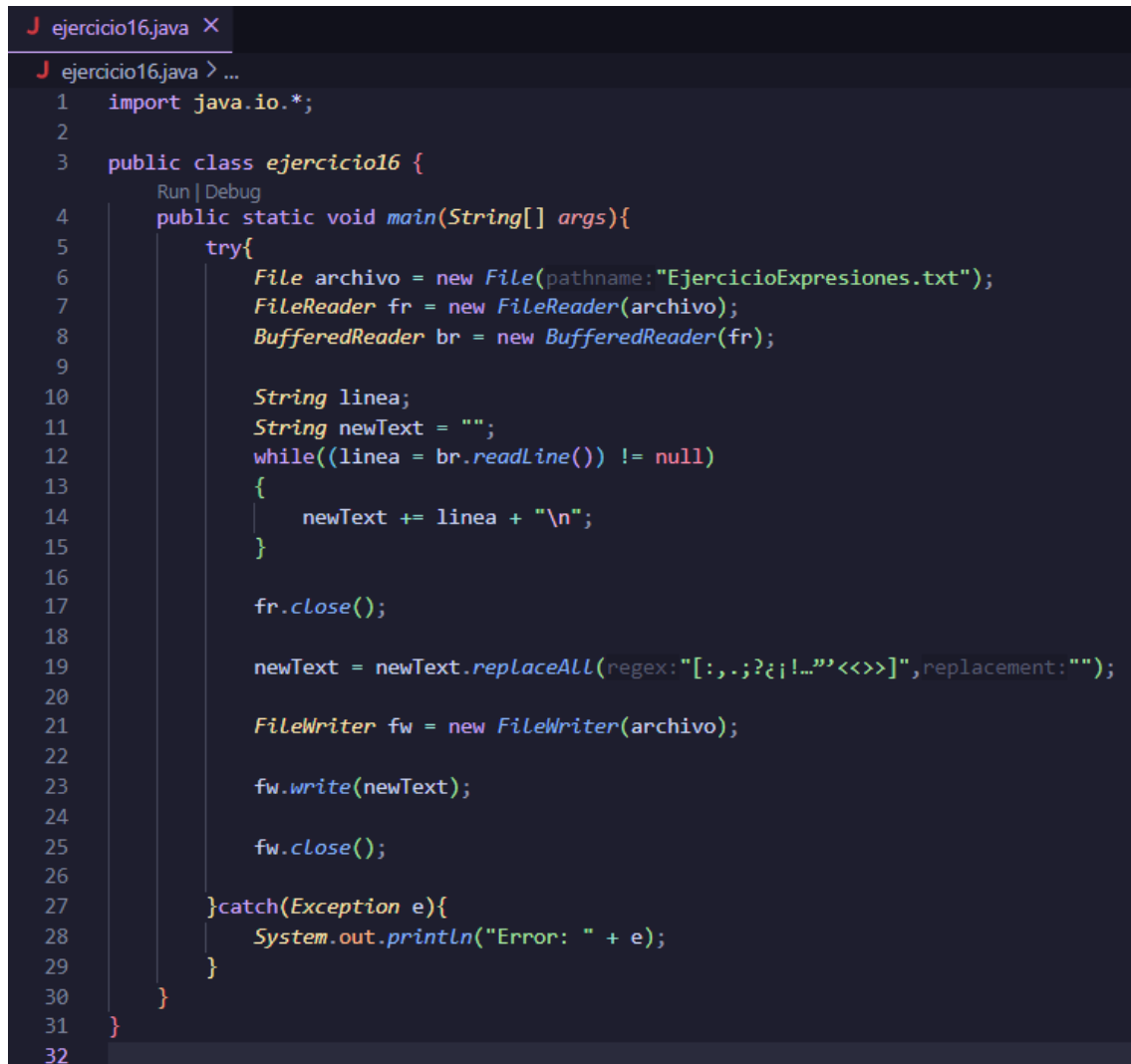
El patrón es diferente al primero pero ninguno de ellos funciona correctamente con el método Split.

Usando el ultimo patrón: “<.*?>(.*)<\V.*?>”, tampoco produce salida.

En resumen, los patrones dados no funcionan para el método Split.

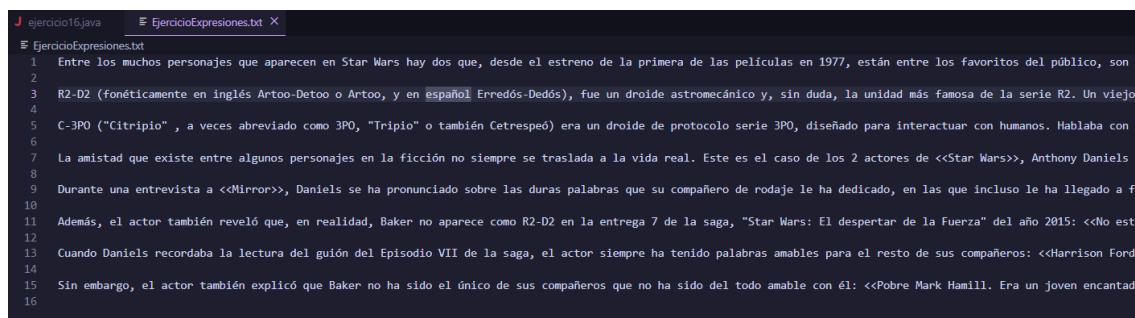
Ejercicio 16. Elimina los símbolos (:,.;?¿¡!...”’<>”) del texto que aparece en el fichero “EjercicioExpresiones.txt”.

Expresión regular usada: “[:.,;?¿¡!...”’<>”]”



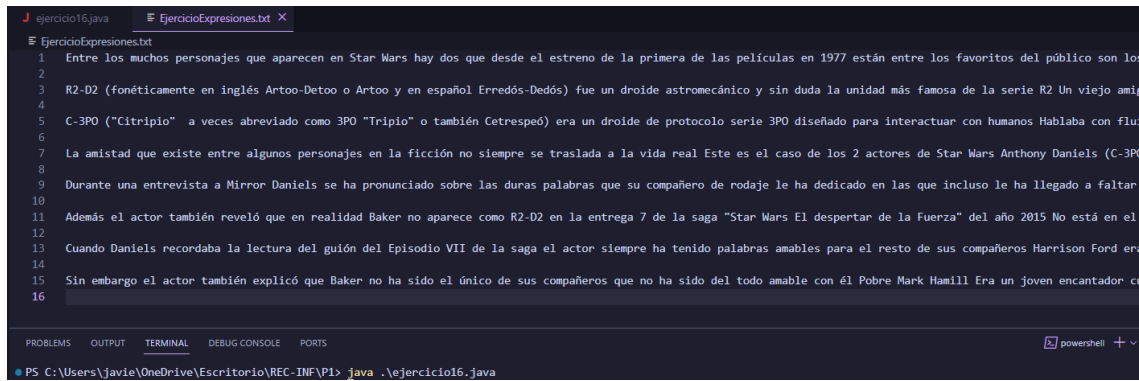
```
ejercicio16.java X
ejercicio16.java > ...
1  import java.io.*;
2
3  public class ejercicio16 {
4      Run | Debug
5      public static void main(String[] args){
6          try{
7              File archivo = new File(pathname:"EjercicioExpresiones.txt");
8              FileReader fr = new FileReader(archivo);
9              BufferedReader br = new BufferedReader(fr);
10
11              String linea;
12              String newText = "";
13              while((linea = br.readLine()) != null)
14              {
15                  newText += linea + "\n";
16              }
17
18              fr.close();
19
20              newText = newText.replaceAll(regex:"[:.,;?¿¡!...”’<>”]",replacement:"");
21
22              FileWriter fw = new FileWriter(archivo);
23
24              fw.write(newText);
25
26              fw.close();
27          }catch(Exception e){
28              System.out.println("Error: " + e);
29          }
30      }
31  }
32
```

Figura 33 Código del ejercicio



```
EjercicioExpresiones.txt
1  Entre los muchos personajes que aparecen en Star Wars hay dos que, desde el estreno de la primera de las películas en 1977, están entre los favoritos del público, son 1
2
3  R2-D2 (fonéticamente en inglés Arttoo-Detoo o Arttoo, y en español Erredós-Dedós), fue un droide astromecánico y, sin duda, la unidad más famosa de la serie R2. Un viejo
4
5  C-3PO ("Citripio" , a veces abreviado como 3PO, "Tripio" o también Cetrespeó) era un droide de protocolo serie 3PO, diseñado para interactuar con humanos. Hablaba con f
6
7  La amistad que existe entre algunos personajes en la ficción no siempre se traslada a la vida real. Este es el caso de los 2 actores de <<Star Wars>>, Anthony Daniels (
8
9  Durante una entrevista a <<Mirror>>, Daniels se ha pronunciado sobre las duras palabras que su compañero de rodaje le ha dedicado, en las que incluso le ha llegado a fa
10
11  Además, el actor también reveló que, en realidad, Baker no aparece como R2-D2 en la entrega 7 de la saga, "Star Wars: El despertar de la Fuerza" del año 2015: <<No está
12
13  Cuando Daniels recordaba la lectura del guión del Episodio VII de la saga, el actor siempre ha tenido palabras amables para el resto de sus compañeros: <<Harrison Ford
14
15  Sin embargo, el actor también explicó que Baker no ha sido el único de sus compañeros que no ha sido del todo amable con él: <<Pobre Mark Hamill. Era un joven encantado
16
```

Figura 34 Fichero antes de la ejecución

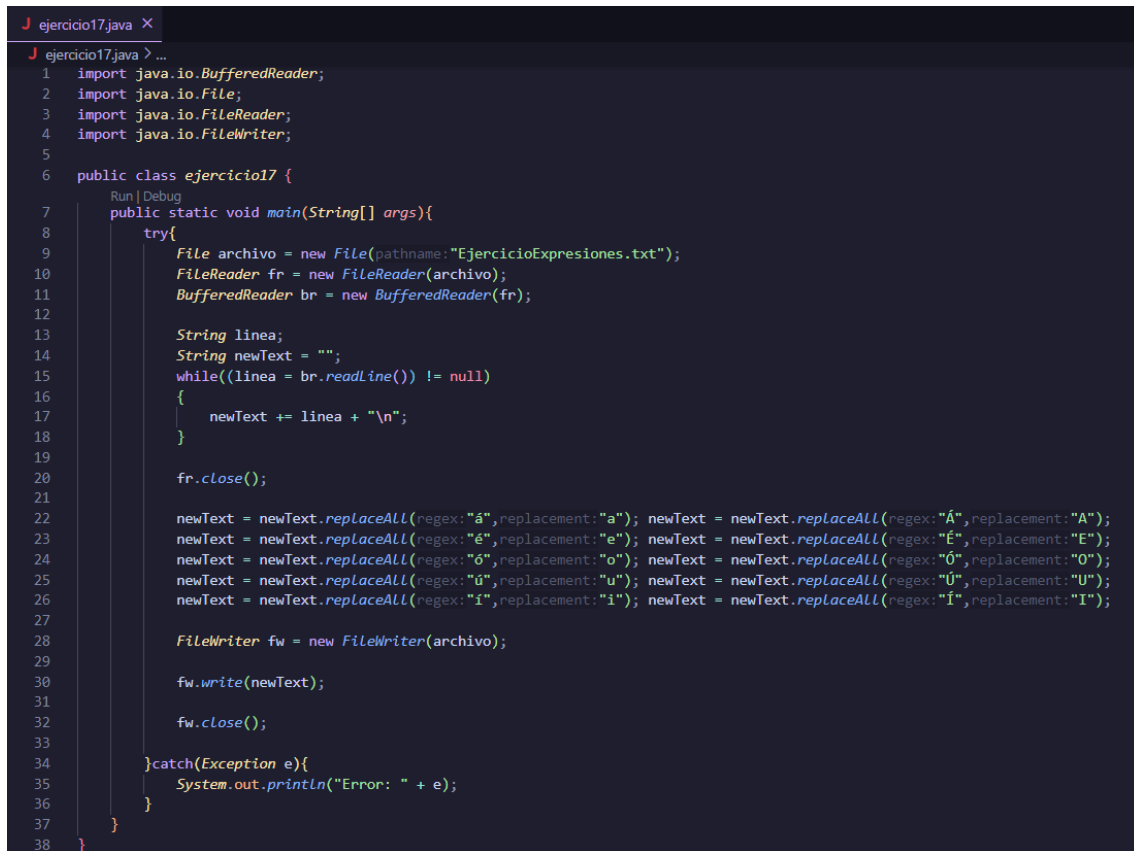


```
ejercicio16.java EjercicioExpresiones.txt
EjercicioExpresiones.txt
1 Entre los muchos personajes que aparecen en Star Wars hay dos que desde el estreno de la primera de las películas en 1977 están entre los favoritos del público son los
2
3 R2-D2 (fonéticamente en inglés Artoo-Detoo o Artoo y en español Erredós-Dedós) fue un droide astromecánico y sin duda la unidad más famosa de la serie R2. Un viejo amigo
4
5 C-3PO ("Citripio" a veces abreviado como 3PO "Tripio" o también Cetrespeó) era un droide de protocolo serie 3PO diseñado para interactuar con humanos. Hablaba con fluidez
6
7 La amistad que existe entre algunos personajes en la ficción no siempre se traslada a la vida real. Este es el caso de los 2 actores de Star Wars: Anthony Daniels (C-3PO) y
8
9 Durante una entrevista a Mirror, Daniels se ha pronunciado sobre las duras palabras que su compañero de rodaje le ha dedicado en las que incluso le ha llegado a faltar
10
11 Además, el actor también reveló que en realidad Baker no aparece como R2-D2 en la entrega 7 de la saga "Star Wars: El despertar de la Fuerza" del año 2015. No está en el
12
13 Cuando Daniels recordaba la lectura del guión del Episodio VII de la saga, el actor siempre ha tenido palabras amables para el resto de sus compañeros. Harrison Ford era
14
15 Sin embargo, el actor también explicó que Baker no ha sido el único de sus compañeros que no ha sido del todo amable con él. El pobre Mark Hamill. Era un joven encantador cuando
16
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\PI> java .\ejercicio16.java
```

Figura 35 Fichero después de la ejecución

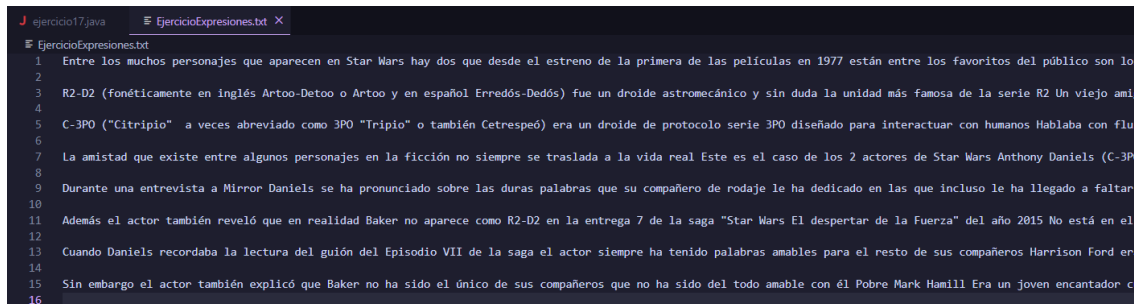
Como podemos ver, los caracteres deseados son removidos con éxito.

Ejercicio 17. Quita las tildes del texto obtenido en el ejercicio anterior; reemplaza por la letra no acentuada.



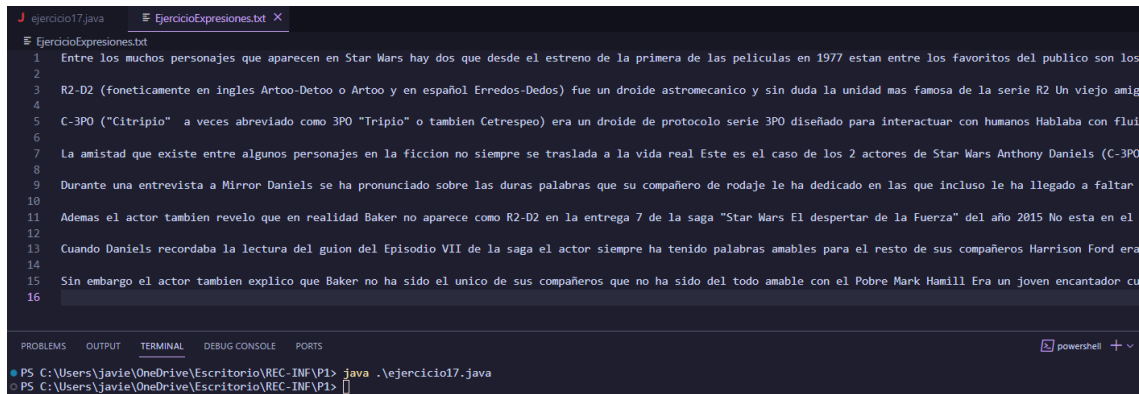
```
J ejercicio17.java X
ejercicio17.java > ...
1 import java.io.BufferedReader;
2 import java.io.File;
3 import java.io.FileReader;
4 import java.io.FileWriter;
5
6 public class ejercicio17 {
7     public static void main(String[] args){
8         try{
9             File archivo = new File(pathname:"EjercicioExpresiones.txt");
10            FileReader fr = new FileReader(archivo);
11            BufferedReader br = new BufferedReader(fr);
12
13            String linea;
14            String newText = "";
15            while((linea = br.readLine()) != null)
16            {
17                newText += linea + "\n";
18            }
19
20            fr.close();
21
22            newText = newText.replaceAll(regex:"á",replacement:"a"); newText = newText.replaceAll(regex:"Á",replacement:"A");
23            newText = newText.replaceAll(regex:"é",replacement:"e"); newText = newText.replaceAll(regex:"É",replacement:"E");
24            newText = newText.replaceAll(regex:"ó",replacement:"o"); newText = newText.replaceAll(regex:"Ó",replacement:"O");
25            newText = newText.replaceAll(regex:"ú",replacement:"u"); newText = newText.replaceAll(regex:"Ú",replacement:"U");
26            newText = newText.replaceAll(regex:"í",replacement:"i"); newText = newText.replaceAll(regex:"Í",replacement:"I");
27
28            FileWriter fw = new FileWriter(archivo);
29
30            fw.write(newText);
31
32            fw.close();
33
34        }catch(Exception e){
35            System.out.println("Error: " + e);
36        }
37    }
38 }
```

Figura 36 Código del ejercicio 17



```
J ejercicio17.java EjercicioExpresiones.txt X
EjercicioExpresiones.txt
1 Entre los muchos personajes que aparecen en Star Wars hay dos que desde el estreno de la primera de las películas en 1977 están entre los favoritos del público son los
2
3 R2-D2 (fonéticamente en inglés Arttoo-Detoo o Arttoo y en español Erredós-Dedós) fue un droide astromecánico y sin duda la unidad más famosa de la serie R2 Un viejo amigo
4
5 C-3PO ("Citripio" a veces abreviado como 3PO "Tripio" o también Cetrespé) era un droide de protocolo serie 3PO diseñado para interactuar con humanos Hablaba con fluidez
6
7 La amistad que existe entre algunos personajes en la ficción no siempre se traslada a la vida real Este es el caso de los 2 actores de Star Wars Anthony Daniels (C-3PO)
8
9 Durante una entrevista a Mirror Daniels se ha pronunciado sobre las duras palabras que su compañero de rodaje le ha dedicado en las que incluso le ha llegado a faltar
10
11 Además el actor también reveló que en realidad Baker no aparece como R2-D2 en la entrega 7 de la saga "Star Wars El despertar de la Fuerza" del año 2015 No está en el
12
13 Cuando Daniels recordaba la lectura del guión del Episodio VII de la saga el actor siempre ha tenido palabras amables para el resto de sus compañeros Harrison Ford era
14
15 Sin embargo el actor también explicó que Baker no ha sido el único de sus compañeros que no ha sido del todo amable con él Pobre Mark Hamill Era un joven encantador
16
```

Figura 37 Fichero antes de la ejecución



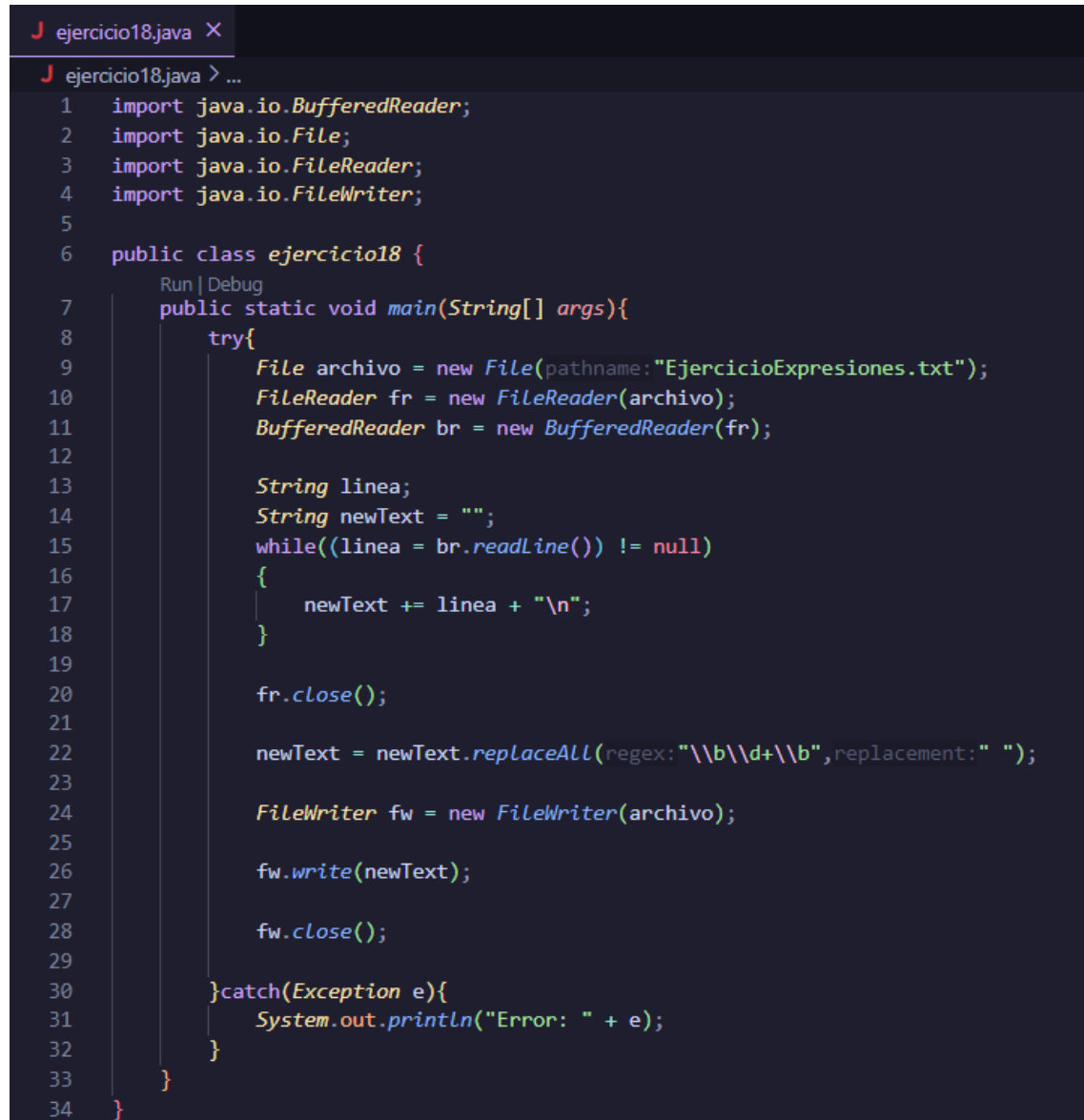
```
ejercicio17.java EjercicioExpresiones.txt X
EjercicioExpresiones.txt
1 Entre los muchos personajes que aparecen en Star Wars hay dos que desde el estreno de la primera de las películas en 1977 están entre los favoritos del público. Son los
2
3 R2-D2 (fonéticamente en inglés Arttoo-Detoo o Arttoo y en español Erredos-Dedos) fue un droide astromecánico y sin duda la unidad más famosa de la serie R2. Un viejo amigo
4
5 C-3PO ("Citripio" a veces abreviado como 3PO "Tripio" o también Cetrespeo) era un droide de protocolo serie 3PO diseñado para interactuar con humanos. Hablaba con fluidez
6
7 La amistad que existe entre algunos personajes en la ficción no siempre se traslada a la vida real. Este es el caso de los 2 actores de Star Wars: Anthony Daniels (C-3PO)
8
9 Durante una entrevista a Mirror, Daniels se ha pronunciado sobre las duras palabras que su compañero de rodaje le ha dedicado en las que incluso le ha llegado a faltar
10
11 Además, el actor también reveló que en realidad Baker no aparece como R2-D2 en la entrega 7 de la saga "Star Wars: El despertar de la Fuerza" del año 2015. No está en el
12
13 Cuando Daniels recordaba la lectura del guion del Episodio VII de la saga, el actor siempre ha tenido palabras amables para el resto de sus compañeros. Harrison Ford era
14
15 Sin embargo, el actor también explicó que Baker no ha sido el único de sus compañeros que no ha sido del todo amable con el pobre Mark Hamill. Era un joven encantador cuando
16
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio17.java
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1>
```

Figura 38 Fichero después de la ejecución

Como podemos ver, los caracteres deseados son sustituidos con éxito.

Ejercicio 18. Reemplaza, del resultado obtenido en el ejercicio anterior, las palabras formadas únicamente por números (ojo, no las palabras con números como H2O, R2-D2, etc) por un espacio.

Expresión regular usada: “\\b\\d+\\b”

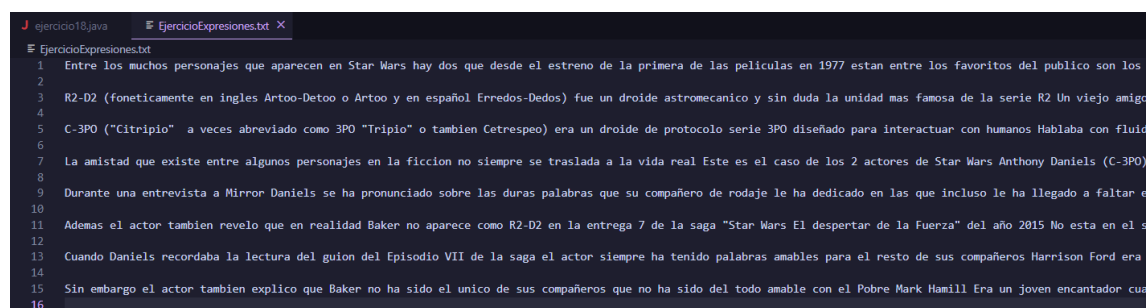


```

1  import java.io.BufferedReader;
2  import java.io.File;
3  import java.io.FileReader;
4  import java.io.FileWriter;
5
6  public class ejercicio18 {
7      public static void main(String[] args){
8          try{
9              File archivo = new File(pathname:"EjercicioExpresiones.txt");
10             FileReader fr = new FileReader(archivo);
11             BufferedReader br = new BufferedReader(fr);
12
13             String linea;
14             String newText = "";
15             while((linea = br.readLine()) != null)
16             {
17                 newText += linea + "\n";
18             }
19
20             fr.close();
21
22             newText = newText.replaceAll(regex:"\\b\\d+\\b",replacement:" ");
23
24             FileWriter fw = new FileWriter(archivo);
25
26             fw.write(newText);
27
28             fw.close();
29         }catch(Exception e){
30             System.out.println("Error: " + e);
31         }
32     }
33 }
34

```

Figura 39 Código del ejercicio 18

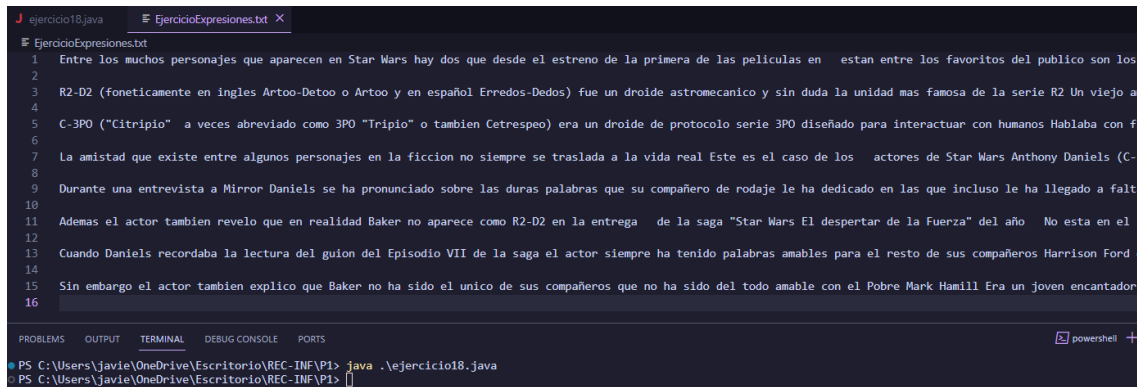


```

1  Entre los muchos personajes que aparecen en Star Wars hay dos que desde el estreno de la primera de las películas en 1977 están entre los favoritos del público son los
2
3  R2-D2 (fonéticamente en inglés Artoo-Detoo o Artoo y en español Erredos-Dedos) fue un droide astromecánico y sin duda la unidad más famosa de la serie R2. Un viejo amigo
4
5  C-3PO ("Citripio" a veces abreviado como 3PO "Tripio" o también Cetrespeo) era un droide de protocolo serie 3PO diseñado para interactuar con humanos. Hablaba con fluidez
6
7  La amistad que existe entre algunos personajes en la ficción no siempre se traslada a la vida real. Este es el caso de los 2 actores de Star Wars: Anthony Daniels (C-3PO)
8
9  Durante una entrevista a Mirror, Daniels se ha pronunciado sobre las duras palabras que su compañero de rodaje le ha dedicado en las que incluso le ha llegado a faltarle
10
11 Además, el actor también reveló que en realidad Baker no aparece como R2-D2 en la entrega 7 de la saga "Star Wars: El despertar de la Fuerza" del año 2015. No está en el
12
13 Cuando Daniels recordaba la lectura del guion del Episodio VII de la saga, el actor siempre ha tenido palabras amables para el resto de sus compañeros. Harrison Ford era
14
15 Sin embargo, el actor también explicó que Baker no ha sido el único de sus compañeros que no ha sido del todo amable con el pobre Mark Hamill. Era un joven encantador cuando
16

```

Figura 40 Fichero antes de la ejecución



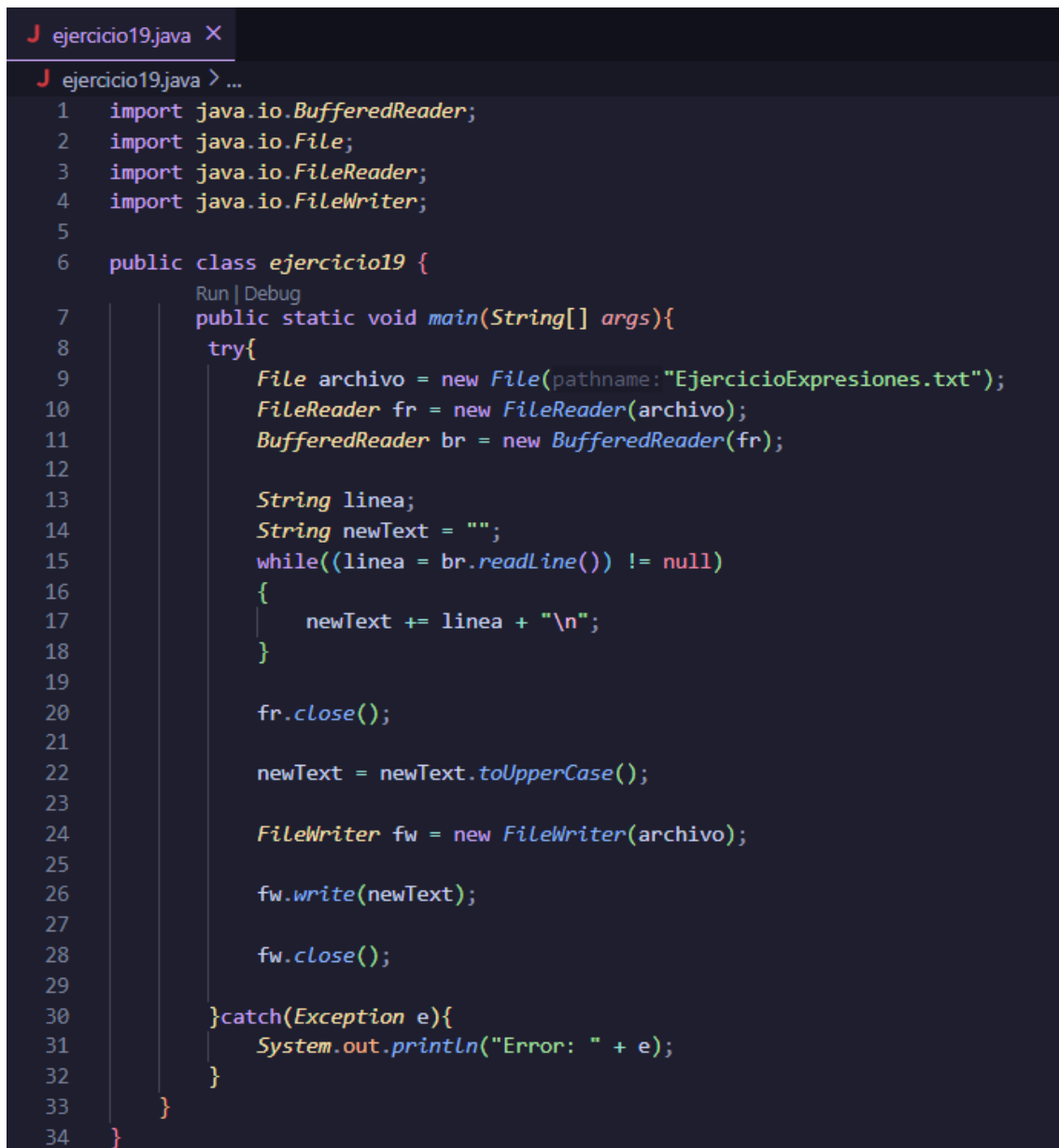
```
ejercicio18.java EjercicioExpresiones.txt X
EjercicioExpresiones.txt
1 Entre los muchos personajes que aparecen en Star Wars hay dos que desde el estreno de la primera de las películas en  están entre los favoritos del público son los
2
3 R2-D2 (fonéticamente en inglés Artoo-Detoo o Artoo y en español Erredos-Dedos) fue un droide astromecánico y sin duda la unidad más famosa de la serie R2. Un viejo am
4
5 C-3PO ("Citripio" a veces abreviado como 3PO "Tripio" o también Cetrespeo) era un droide de protocolo serie 3PO diseñado para interactuar con humanos. Hablaba con fl
6
7 La amistad que existe entre algunos personajes en la ficción no siempre se traslada a la vida real. Este es el caso de los  actores de Star Wars Anthony Daniels (C-3
8
9 Durante una entrevista a Mirror, Daniels se ha pronunciado sobre las duras palabras que su compañero de rodaje le ha dedicado en las que incluso le ha llegado a falta
10
11 Además, el actor también reveló que en realidad Baker no aparece como R2-D2 en la entrega  de la saga "Star Wars: El despertar de la Fuerza" del año  . No está en el s
12
13 Cuando Daniels recordaba la lectura del guion del Episodio VII de la saga, el actor siempre ha tenido palabras amables para el resto de sus compañeros Harrison Ford e
14
15 Sin embargo, el actor también explicó que Baker no ha sido el único de sus compañeros que no ha sido del todo amable con el pobre Mark Hamill. Era un joven encantador
16

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\PI> java .\ejercicio18.java
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\PI>
```

Figura 41 Fichero después de la ejecución

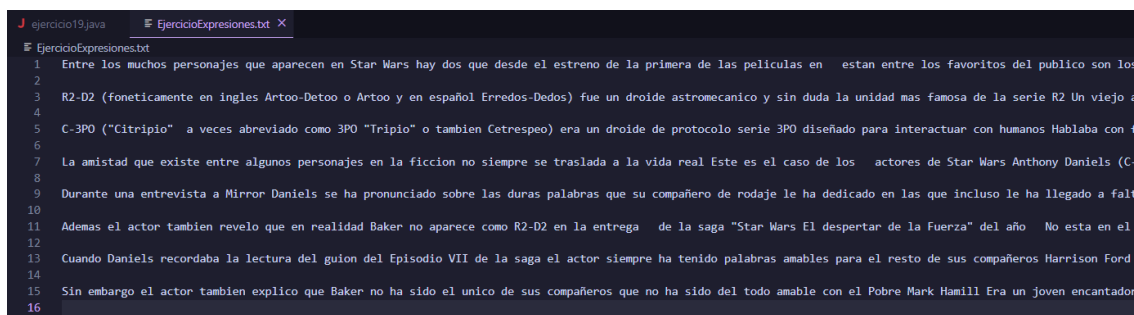
Como podemos ver, las palabras deseadas son sustituidas por un espacio.

Ejercicio 19. Convierte el texto anterior a mayúsculas



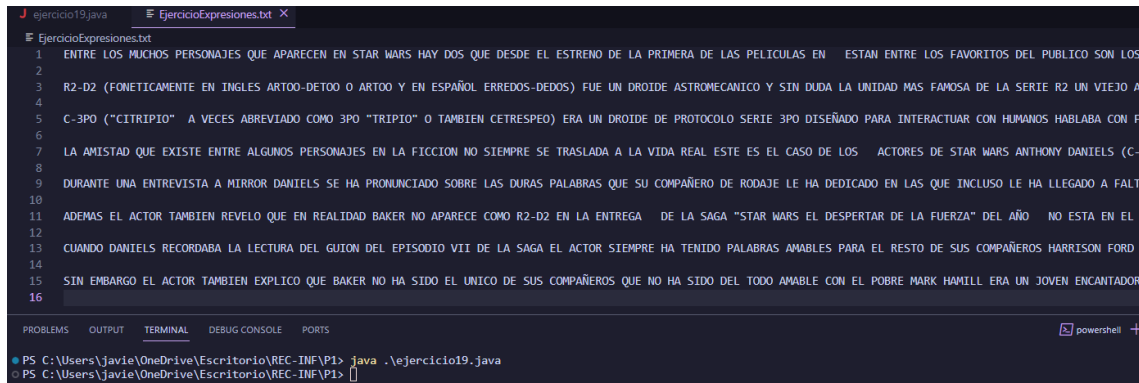
```
ejercicio19.java X
ejercicio19.java > ...
1  import java.io.BufferedReader;
2  import java.io.File;
3  import java.io.FileReader;
4  import java.io.FileWriter;
5
6  public class ejercicio19 {
7      Run | Debug
8      try{
9          File archivo = new File(pathname:"EjercicioExpresiones.txt");
10         FileReader fr = new FileReader(archivo);
11         BufferedReader br = new BufferedReader(fr);
12
13         String linea;
14         String newText = "";
15         while((linea = br.readLine()) != null)
16         {
17             newText += linea + "\n";
18         }
19
20         fr.close();
21
22         newText = newText.toUpperCase();
23
24         FileWriter fw = new FileWriter(archivo);
25
26         fw.write(newText);
27
28         fw.close();
29     }catch(Exception e){
30         System.out.println("Error: " + e);
31     }
32 }
33
34 }
```

Figura 42 Código ejercicio 19



```
ejercicio19.java  EjercicioExpresiones.txt X
EjercicioExpresiones.txt
1  Entre los muchos personajes que aparecen en Star Wars hay dos que desde el estreno de la primera de las películas en  están entre los favoritos del público son los
2
3  R2-D2 (fonéticamente en inglés Artoo-Detoo o Artoo y en español Erredos-Dedos) fue un droide astromecánico y sin duda la unidad más famosa de la serie R2 Un viejo am
4
5  C-3PO ("Citripio" a veces abreviado como 3PO "Tripio" o también Cetrespeo) era un droide de protocolo serie 3PO diseñado para interactuar con humanos Hablaba con fl
6
7  La amistad que existe entre algunos personajes en la ficción no siempre se traslada a la vida real Este es el caso de los  actores de Star Wars Anthony Daniels (C-3
8
9  Durante una entrevista a Mirror Daniels se ha pronunciado sobre las duras palabras que su compañero de rodaje le ha dedicado en las que incluso le ha llegado a falta
10
11 Además el actor también reveló que en realidad Baker no aparece como R2-D2 en la entrega  de la saga "Star Wars El despertar de la Fuerza" del año  No está en el s
12
13 Cuando Daniels recordaba la lectura del guion del Episodio VII de la saga el actor siempre ha tenido palabras amables para el resto de sus compañeros Harrison Ford e
14
15 Sin embargo el actor también explicó que Baker no ha sido el único de sus compañeros que no ha sido del todo amable con el Pobre Mark Hamill Era un joven encantador
16
```

Figura 43 Fichero antes de la ejecución



```
ejercicio19.java EjercicioExpresiones.txt X
EjercicioExpresiones.txt
1 ENTRE LOS MUCHOS PERSONAJES QUE APARECEN EN STAR WARS HAY DOS QUE DESDE EL ESTRENO DE LA PRIMERA DE LAS PELICULAS EN ESTAN ENTRE LOS FAVORITOS DEL PUBLICO SON LOS
2
3 R2-D2 (FONETICAMENTE EN INGLES ARTOO-DETOO O ARTOO Y EN ESPAÑOL ERREDOS-DEDS) FUE UN DROIDE ASTROMECHANICO Y SIN DUDA LA UNIDAD MAS FAMOSA DE LA SERIE R2 UN VIEJO A
4
5 C-3PO ("CITRIPIO" A VECES ABBREVIADO COMO 3PO "TRIPPIO" O TAMBIEN CETRESPEO) ERA UN DROIDE DE PROTOCOLO SERIE 3PO DISEÑADO PARA INTERACTUAR CON HUMANOS HABLABA CON F
6
7 LA AMISTAD QUE EXISTE ENTRE ALGUNOS PERSONAJES EN LA FICCION NO SIEMPRE SE TRASLADA A LA VIDA REAL ESTE ES EL CASO DE LOS ACTORES DE STAR WARS ANTHONY DANIELS (C-
8
9 DURANTE UNA ENTREVISTA A MIRROR DANIELS SE HA PRONUNCIADO SOBRE LAS DURAS PALABRAS QUE SU COMPAÑERO DE RODAJE LE HA DEDICADO EN LAS QUE INCLUSO LE HA LLEGADO A FALT
10
11 ADEMAS EL ACTOR TAMBIEN REVELO QUE EN REALIDAD BAKER NO APARECE COMO R2-D2 EN LA ENTREGA DE LA SAGA "STAR WARS EL DESPERTAR DE LA FUERZA" DEL AÑO NO ESTA EN EL S
12
13 CUANDO DANIELS RECORDABA LA LECTURA DEL GUION DEL EPISODIO VII DE LA SAGA EL ACTOR SIEMPRE HA TENIDO PALABRAS AMABLES PARA EL RESTO DE SUS COMPAÑEROS HARRISON FORD
14
15 SIN EMBARGO EL ACTOR TAMBIEN EXPLICO QUE BAKER NO HA SIDO EL UNICO DE SUS COMPAÑEROS QUE NO HA SIDO DEL TODO AMABLE CON EL POBRE MARK HAMILL ERA UN JOVEN ENCANTADOR
16

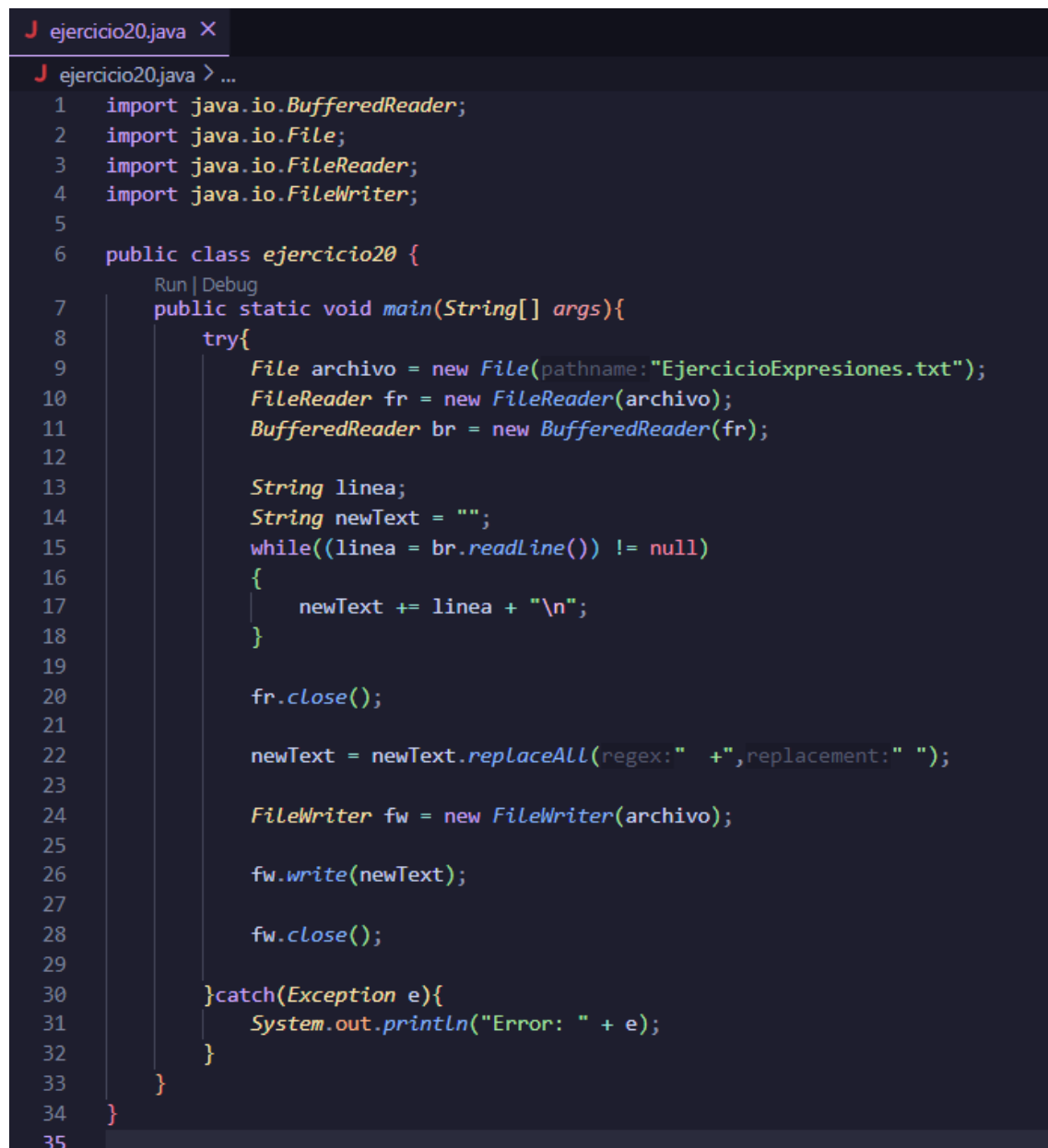
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio19.java
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> []
```

Figura 44 Fichero después de la ejecución

Como podemos ver, con el método `toUpperCase` hacemos que todas las letras se reemplacen por su versión en mayúscula.

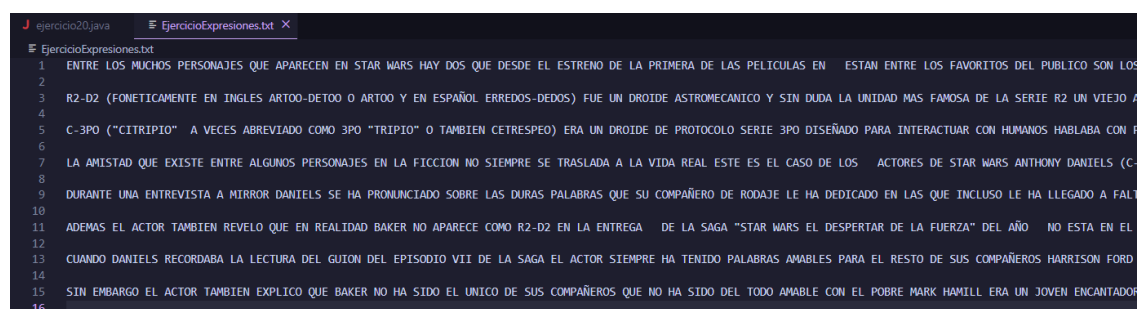
Ejercicio 20. Reemplaza los dobles espacios que se hayan podido crear por un único espacio.

Expresión regular usada: “ +”



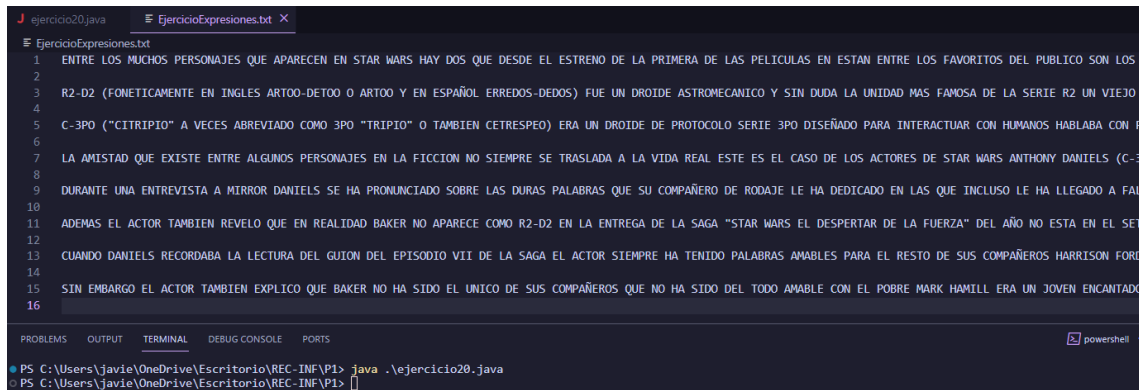
```
1  import java.io.BufferedReader;
2  import java.io.File;
3  import java.io.FileReader;
4  import java.io.FileWriter;
5
6  public class ejercicio20 {
7      Run | Debug
8      public static void main(String[] args){
9          try{
10             File archivo = new File(pathname:"EjercicioExpresiones.txt");
11             FileReader fr = new FileReader(archivo);
12             BufferedReader br = new BufferedReader(fr);
13
14             String linea;
15             String newText = "";
16             while((linea = br.readLine()) != null)
17             {
18                 newText += linea + " ";
19             }
20
21             fr.close();
22
23             newText = newText.replaceAll(regex:" +",replacement:" ");
24
25             FileWriter fw = new FileWriter(archivo);
26
27             fw.write(newText);
28
29             fw.close();
30
31         }catch(Exception e){
32             System.out.println("Error: " + e);
33         }
34     }
35 }
```

Figura 45 Código del ejercicio 20



```
1  ENTRE LOS MUCHOS PERSONAJES QUE APARECEN EN STAR WARS HAY DOS QUE DESDE EL ESTRENO DE LA PRIMERA DE LAS PELICULAS EN ESTAN ENTRE LOS FAVORITOS DEL PUBLICO SON LOS
2
3  R2-D2 (FONETICAMENTE EN INGLES ARTOO-DETOO O ARTOO Y EN ESPAÑOL ERREDOS-DEDOS) FUE UN DROIDE ASTROMECANICO Y SIN DUDA LA UNIDAD MAS FAMOSA DE LA SERIE R2 UN VIEJO AM
4
5  C-3PO ("CITRIPIO" A VECES ABBREVIADO COMO 3PO "TRIPPIO" O TAMBIEN CETRESPEO) ERA UN DROIDE DE PROTOCOLO SERIE 3PO DISEÑADO PARA INTERACTUAR CON HUMANOS HABLABA CON FL
6
7  LA AMISTAD QUE EXISTE ENTRE ALGUNOS PERSONAJES EN LA FICCION NO SIEMPRE SE TRASLADA A LA VIDA REAL ESTE ES EL CASO DE LOS ACTORES DE STAR WARS ANTHONY DANIELS (C-3
8
9  DURANTE UNA ENTREVISTA A MIRROR DANIELS SE HA PRONUNCIADO SOBRE LAS DURAS PALABRAS QUE SU COMPAÑERO DE RODAJE LE HA DEDICADO EN LAS QUE INCLUSO LE HA LLEGADO A FALTA
10
11 ADEMAS EL ACTOR TAMBIEN REVELO QUE EN REALIDAD BAKER NO APARECE COMO R2-D2 EN LA ENTREGA DE LA SAGA "STAR WARS EL DESPERTAR DE LA FUERZA" DEL AÑO NO ESTA EN EL S
12
13 CUANDO DANIELS RECORDABA LA LECTURA DEL GUILON DEL EPISODIO VII DE LA SAGA EL ACTOR SIEMPRE HA TENIDO PALABRAS AMABLES PARA EL RESTO DE SUS COMPAÑEROS HARRISON FORD E
14
15 SIN EMBARGO EL ACTOR TAMBIEN EXPLICO QUE BAKER NO HA SIDO EL UNICO DE SUS COMPAÑEROS QUE NO HA SIDO DEL TODO AMABLE CON EL POBRE MARK HAMILL ERA UN JOVEN ENCANTADOR
16
```

Figura 46 Fichero antes de la ejecución



```
ejercicio20.java EjercicioExpresiones.txt
1 ENTRE LOS MUCHOS PERSONAJES QUE APARECEN EN STAR WARS HAY DOS QUE DESDE EL ESTRENO DE LA PRIMERA DE LAS PELICULAS EN ESTAN ENTRE LOS FAVORITOS DEL PUBLICO SON LOS
2
3 R2-D2 (FONETICAMENTE EN INGLES ARTOO-DETOO O ARTOO Y EN ESPAÑOL ERREDOS-DEDOS) FUE UN DROIDE ASTROMECHANICO Y SIN DUDA LA UNIDAD MAS FAMOSA DE LA SERIE R2 UN VIEJO
4
5 C-3PO ("CITRIPIO" A VECES ABREVIADO COMO 3PO "TRIPPIO" O TAMBIEN CETRESPEO) ERA UN DROIDE DE PROTOCOLO SERIE 3PO DISEÑADO PARA INTERACTUAR CON HUMANOS HABLABA CON F
6
7 LA AMISTAD QUE EXISTE ENTRE ALGUNOS PERSONAJES EN LA FICCION NO SIEMPRE SE TRASLADA A LA VIDA REAL ESTE ES EL CASO DE LOS ACTORES DE STAR WARS ANTHONY DANIELS (C-3
8
9 DURANTE UNA ENTREVISTA A MIRROR DANIELS SE HA PRONUNCIADO SOBRE LAS DURAS PALABRAS QUE SU COMPAÑERO DE RODAJE LE HA DEDICADO EN LAS QUE INCLUSO LE HA LLEGADO A FAL
10
11 ADEMAS EL ACTOR TAMBIEN REVELO QUE EN REALIDAD BAKER NO APARECE COMO R2-D2 EN LA ENTREGA DE LA SAGA "STAR WARS EL DESPERTAR DE LA FUERZA" DEL AÑO NO ESTA EN EL SET
12
13 CUANDO DANIELS RECORDABA LA LECTURA DEL GUION DEL EPISODIO VII DE LA SAGA EL ACTOR SIEMPRE HA TENIDO PALABRAS AMABLES PARA EL RESTO DE SUS COMPAÑEROS HARRISON FORD
14
15 SIN EMBARGO EL ACTOR TAMBIEN EXPLICO QUE BAKER NO HA SIDO EL UNICO DE SUS COMPAÑEROS QUE NO HA SIDO DEL TODO AMABLE CON EL POBRE MARK HAMILL ERA UN JOVEN ENCANTADO
16

PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE PORTS
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> java .\ejercicio20.java
PS C:\Users\javie\OneDrive\Escritorio\REC-INF\P1> []
```

Figura 47 Fichero después de la ejecución

Como podemos ver, los dobles (o más) espacios, son sustituidos por uno solo.