

Ex 1.

```
Ex1();  
//Ex2(); //Tamb
```

Este ejercicio tiene datos fijos, por lo que no hace falta indicar nada en la ejecución.

```
C:\Users\shote\.jdk\openjdk-23.0.1\bin\java.exe  
Archivo creado: test1.txt
```

```
test1.txt  
test1.txt
```

Al lanzarlo este nos crea el archivo test1.txt, si ya existe lo indica y no genera archivo.

```
C:\Users\shote\.jdk\openjdk-23.0.1\bin\java.exe  
El archivo test1.txt ya existe.
```

Ex 2 y 3.

```
//Ex2();  
Ex2(); //Tambien incluye el Ex3;
```

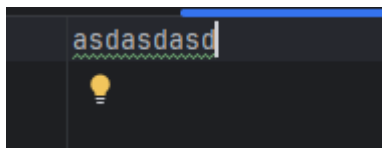
No hace falta ejecutar con nada, ya que los datos son fijos.

```
Archivo creado.  
Nombre: ejemplo1.txt  
Ruta absoluta: D:\Programacion\1DAM\P3ManejoDeFicheros\P3\src\main\resources\ejemplo1.txt  
Longitud: 0  
La carpeta ya existe.
```

Cuando lo lanzamos crea el archivo ejemplo1.txt y nos da detalles de este, si ya existe lo indica.

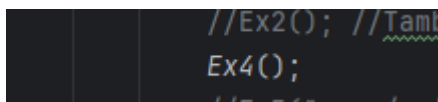
```
El fichero ya existe.  
Nombre: ejemplo1.txt  
Ruta absoluta: D:\Programacion\1DAM\P3ManejoDeFicheros\P3\src\main\resources\ejemplo1.txt  
Longitud: 0  
La carpeta ya existe.
```

Si cambiamos el contenido del archivo esto lo detecta, porque no lo estamos generando de nuevo sino verificando el estado de este mismo.



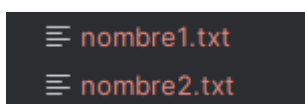
```
C:\Users\shote\.jdk\openjdk-23.0.1\bin\java.exe "-javaagent:D:\Descargas\IntelliJ IDEA Commu
El fichero ya existe.
Nombre: ejemplo1.txt
Ruta absoluta: D:\Programacion\1DAM\P3ManejoDeFicheros\P3\src\main\resources\ejemplo1.txt
Longitud: 9
La carpeta ya existe.
```

Ex 4.

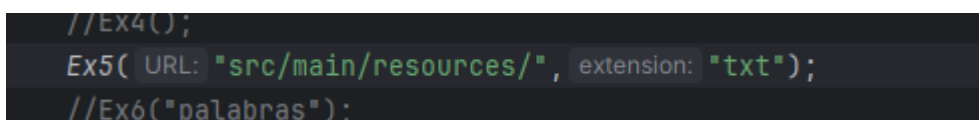


Para esta ejecución no se solicita nada ya que se pide por pantalla. Nos pide ruta y cantidad de archivos a crear. Después los genera.

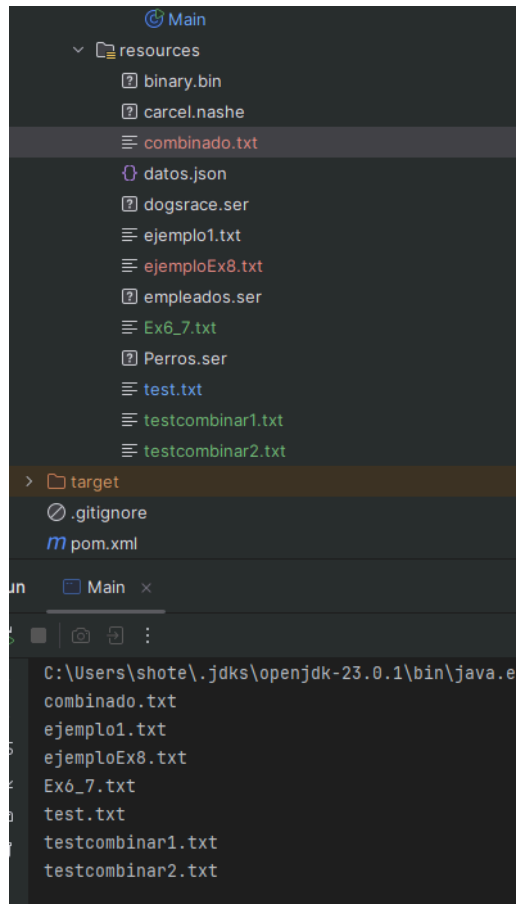
```
C:\Users\shote\.jdk\openjdk-23.0.1\bin\java.exe "-javaagent:D:\Descargas\IntelliJ IDEA Commu
Introduce la carpeta en la que quieres introducir los archivos:
D:\Programacion\1DAM\P3ManejoDeFicheros\P3\src\main\resources
Introduce la catidad de archivos a introducir:
2
Archivo creado: nombre1.txt
Archivo creado: nombre2.txt
```



Ex 5.



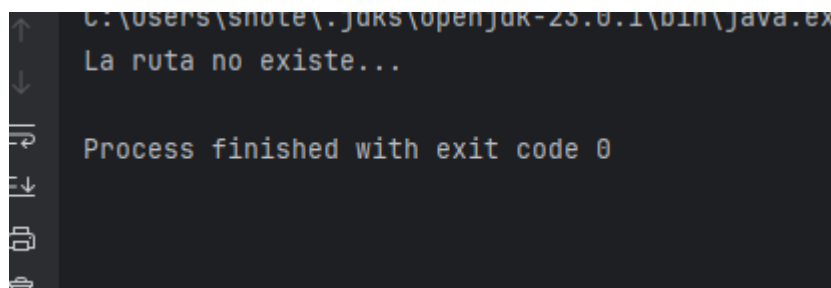
En este ejercicio indicamos la ruta en la cual queremos buscar y la extensión que se quiere buscar antes de ejecutar.



Y nos muestro todos los archivos con esa extensión , en esa ruta en concreto.



Si la ruta no existe, simplemente lo indica por una comprobación del código.



Ex 6 y 7.

```
//Ex5("pdf");  
Ex6( word: "palabras");  
Ex7( word: "palabra");  
//Ex8();
```

Aquí indicamos las palabras por las que queremos buscar.

```
C:\Users\snote\.junks\openjdk-23.0.1\bin\java.exe -javaagent:D:\descargas\IntelliJ IDE  
Dame la ruta del archivo para buscar la palabra:  
D:\Programacion\1DAM\P3ManejoDeFicheros\P3\src\main\resources\Ex6_7.txt  
La palabra palabras aparece: 3 veces.  
Dame la ruta del archivo para buscar la palabra:  
D:\Programacion\1DAM\P3ManejoDeFicheros\P3\src\main\resources\Ex6_7.txt  
La palabra palabra aparece: 2 veces.
```

Aquí pondremos la ruta del archivo y nos devolverá el número de veces que aparece la palabra filtrada. Ambos ejercicios hacen lo mismo pero usando distintas formas / clases.

```
Current File ▾  
Main.java Ex6_7.txt x  
1 Este test busca las palabras con palabras y muchas palabras Si me gusta la palabra palabra lalala lelele lululululu
```

Archivo .txt.

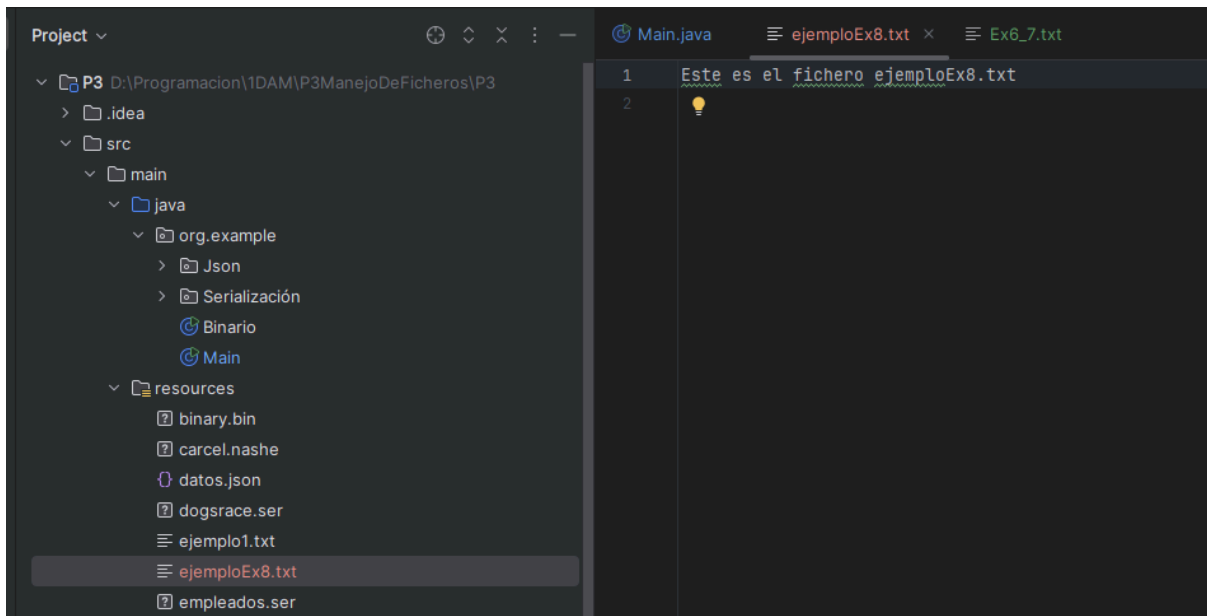
Ex 8.

```
//Ex6("palabras");  
//Ex7("palabra");  
Ex8();  
//Ex9("src/main/resources  
//Ex10("src/main/resource
```

En este script / programa no se solicita ninguna entrada para ejecutar.

```
C:\Users\shote\.jdk\openjdk-23.0.1\bin\java.exe
Archivo creado

Process finished with exit code 0
```



Y lo que hace al ejecutar es crearnos un archivo que indicamos fijamente en el programa. Y ingresa una línea dentro de este que pone Este es el fichero (n) .txt.

```
C:\Users\shote\.jdk\openjdk-23.0.1\bin\java.exe
El archivo ya existe.

Process finished with exit code 0
```

Si el archivo ya existe lo indica.

Ex 9.

```
//Ex8();  
Ex9( filePath: "src/main/resources/test.txt");  
//Ex10("src/main/resources/testcombinar1.txt", "src/
```

Antes de ejecutar, debemos indicar el archivo al cual vamos a afectar, este programa pone la primera letra de cada palabra en mayúsculas.

Antes:

```
1 hola que tal, esto es una prueba para el ejercicio 9.  
2
```

```
C:\Users\shote\.jdk\openjdk-23.0.1\bin\java.exe "-jav  
Archivo modificado correctamente.  
  
Process finished with exit code 0
```

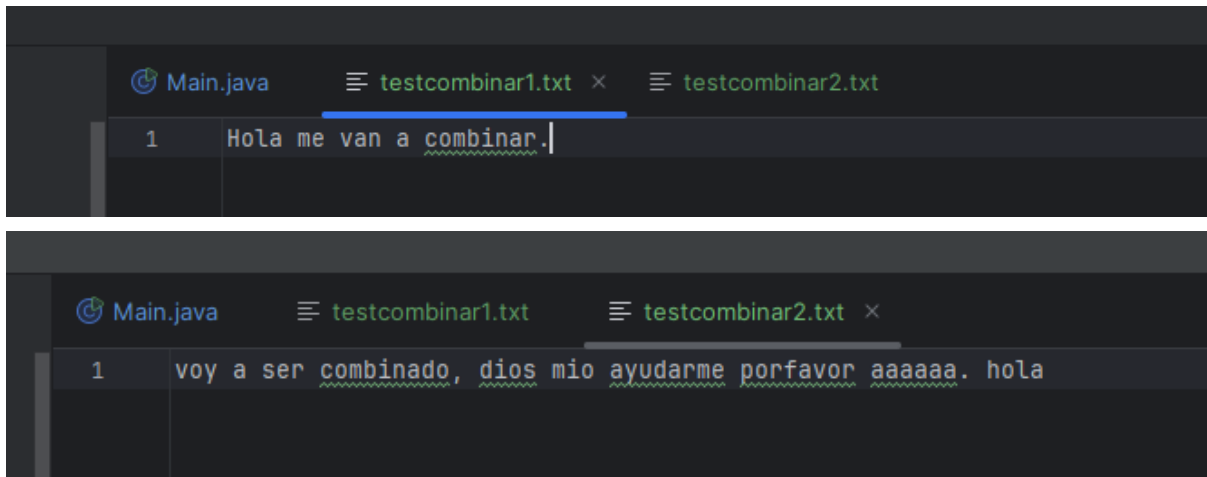
Después:

```
1 Hola Que Tal, Esto Es Una Prueba Para El Ejercicio 9.  
2
```

Ex 10.

```
//Ex9( filePath: "src/main/resources/test.txt");  
Ex10( file1: "src/main/resources/testcombinar1.txt", file2: "src/main/resources/testcombinar2.txt", newFile: "src/main/resources/combinado.txt");
```

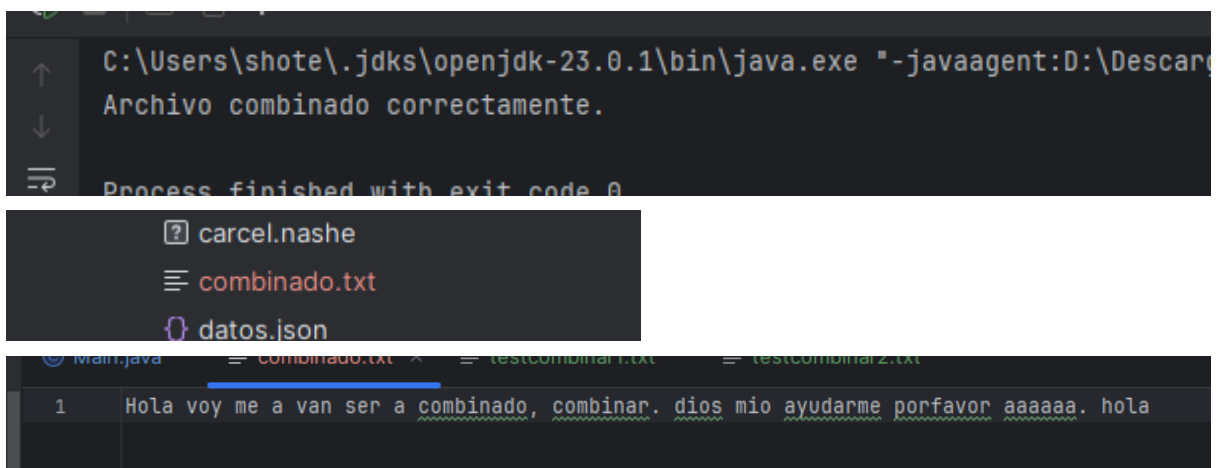
En este ejercicio , lo que buscamos es crear un archivo. Mediante dos existentes con las palabras cruzadas entre ellas. Para la entrada vamos a indicar la ruta de ambos archivos que vamos a combinar y el archivo resultado.



```
Main.java testcombinar1.txt testcombinar2.txt
1 Hola me van a combinar.
```

```
Main.java testcombinar1.txt testcombinar2.txt
1 voy a ser combinado, dios mio ayudarme porfavor aaaaaa. hola
```

Resultado de ejecutar:



```
C:\Users\shote\.jdk\openjdk-23.0.1\bin\java.exe -javaagent:D:\Descarga\carcel.nashe
Archivo combinado correctamente.
Process finished with exit code 0
```

```
carcel.nashe
combinado.txt
datos.json
```

```
Main.java combinado.txt testcombinar1.txt testcombinar2.txt
1 Hola voy me a van ser a combinado, combinar. dios mio ayudarme porfavor aaaaaa. hola
```

Como vemos se nos han cruzado las palabras , y como uno es más largo que otro al final es legible como el original.