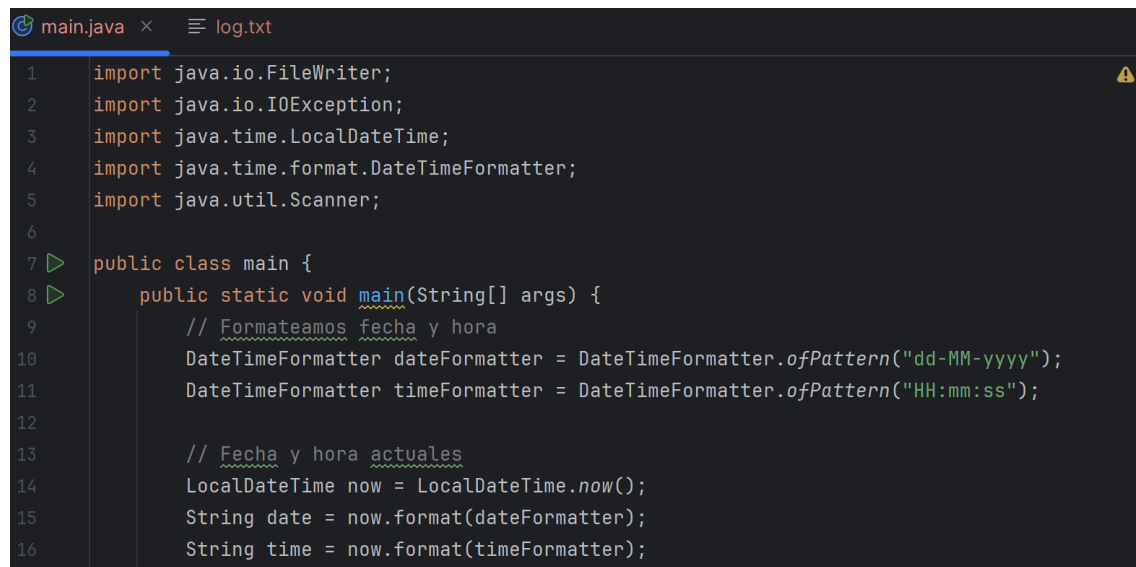


MEMORIA EJERCICIO 01.05

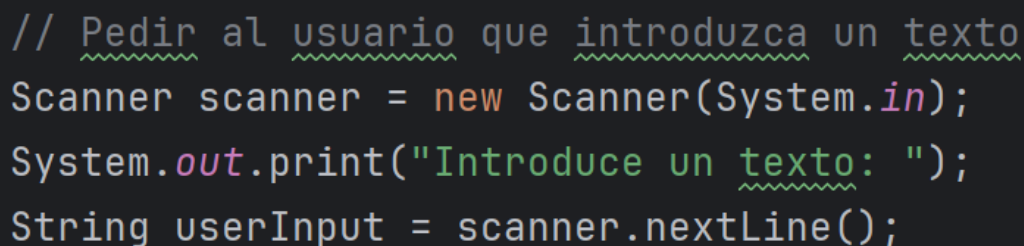
El programa tiene como finalidad registrar en un fichero de texto llamado *log.txt* información cada vez que se ejecuta. Lo que se almacena es la fecha, la hora y el texto que introduce el usuario en la consola. De esta manera, el archivo va guardando un historial con todas las ejecuciones del programa, sin borrar lo anterior, lo que permite llevar un control cronológico de las entradas.

Para conseguirlo, en primer lugar, se importan diferentes librerías de Java. Con `FileWriter` y `IOException` se trabaja la escritura en ficheros y la gestión de posibles errores. Con `LocalDateTime` y `DateTimeFormatter` se obtienen y formatean la fecha y la hora en un estilo más legible, por ejemplo 29-09-2025 para la fecha y 10:23:11 para la hora. Finalmente, se importa `Scanner`, que es la herramienta que permite recoger el texto que el usuario introduce por teclado.



```
1 import java.io.FileWriter;
2 import java.io.IOException;
3 import java.time.LocalDateTime;
4 import java.time.format.DateTimeFormatter;
5 import java.util.Scanner;
6
7 public class main {
8     public static void main(String[] args) {
9         // Formateamos fecha y hora
10        DateTimeFormatter dateFormatter = DateTimeFormatter.ofPattern("dd-MM-yyyy");
11        DateTimeFormatter timeFormatter = DateTimeFormatter.ofPattern("HH:mm:ss");
12
13        // Fecha y hora actuales
14        LocalDateTime now = LocalDateTime.now();
15        String date = now.format(dateFormatter);
16        String time = now.format(timeFormatter);
```

El flujo del programa comienza creando dos formatos: uno para la fecha y otro para la hora. Después, se obtiene la fecha y hora actuales mediante `LocalDateTime.now()` y se guardan como cadenas de texto ya formateadas. A continuación, se pide al usuario que escriba un texto en la consola, el cual se almacena en una variable para poder añadirlo posteriormente al fichero.



```
// Pedir al usuario que introduzca un texto
Scanner scanner = new Scanner(System.in);
System.out.print("Introduce un texto: ");
String userInput = scanner.nextLine();
```

La parte más importante del código es la escritura en *log.txt*. Para ello, se utiliza la clase `FileWriter` con el modo de añadido activado (mediante el parámetro `true`). Esto significa que

cada nueva línea se incorpora al final del fichero sin eliminar lo que ya había. Así, en cada ejecución se va construyendo un registro con la estructura “fecha hora texto”. Por ejemplo, si el usuario introduce la palabra *hola*, el resultado en el archivo podría ser: 29-09-2025 10:45:02 hola. Cada nueva ejecución añadirá otra línea similar debajo.

```
// Crear o añadir al fichero log.txt
try (FileWriter writer = new FileWriter( fileName: "log.txt", append: true)) {
    writer.write( str: date + " " + time + " " + userInput + System.lineSeparator());
    System.out.println("Texto añadido al fichero log.txt");
} catch (IOException e) {
    System.err.println("Error al escribir en el fichero: " + e.getMessage());
}
```

El programa también está protegido contra posibles errores durante la escritura. Todo el proceso de creación o edición del fichero está incluido dentro de un bloque try-catch. En caso de que surja un problema, como la imposibilidad de acceder al archivo, se muestra un mensaje de error por consola indicando la causa.

En conclusión, este programa permite mantener un historial ordenado y creciente de los mensajes introducidos por el usuario, siempre acompañados por la fecha y hora exactas en que se escribieron. Gracias al uso del modo de añadido, el archivo nunca se sobrescribe, lo que lo convierte en una pequeña herramienta de registro o bitácora útil y sencilla.

