

Actividad Práctica - Tema 4: Acceso Aleatorio a Ficheros

Iris Pérez Aparicio
2º DAM
Campus FP Emprende Humanes

AccesoAleatorioEjercicio.java

```
import java.io.*;

public class AccesoAleatorioEjercicio {
    public static void main(String[] args) {
        try {
            // Escribir registros
            RandomAccessFile raf = new
RandomAccessFile("datos/registros.dat", "rw");
            raf.writeUTF("Registro 1");
            raf.writeUTF("Registro 2");
            raf.writeUTF("Registro 3");
            raf.seek(0); // Volver al inicio

            // Leer primer registro
            System.out.println("Posición antes de leer 1: " +
raf.getFilePointer());
            String r1 = raf.readUTF();
            System.out.println("Registro 1: " + r1);
            System.out.println("Posición después de leer 1: " +
raf.getFilePointer());

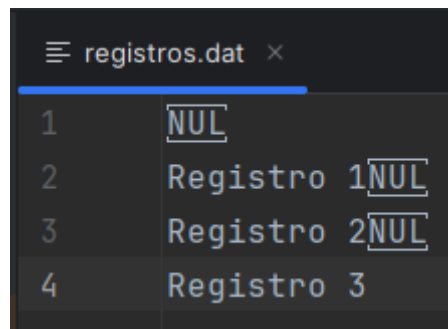
            // Leer segundo registro
            System.out.println("Posición antes de leer 2: " +
raf.getFilePointer());
            String r2 = raf.readUTF();
            System.out.println("Registro 2: " + r2);
            System.out.println("Posición después de leer 2: " +
raf.getFilePointer());
            raf.close();
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

Este programa escribe tres registros con texto en un archivo binario usando writeUTF. Muestra la posición antes de leer el primer registro, lee el registro, lo muestra y luego vuelve a mostrar la posición. Hace exactamente lo mismo con el segundo registro y finalmente cierra el puntero.

EJECUCIÓN

```
Posición antes de leer 1: 0
Registro 1: Registro 1
Posición después de leer 1: 12
Posición antes de leer 2: 12
Registro 2: Registro 2
Posición después de leer 2: 24

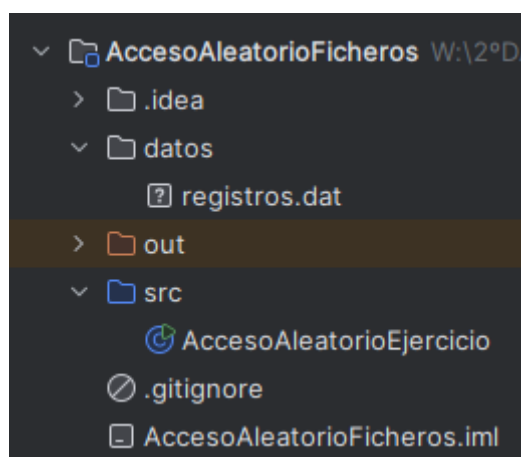
Process finished with exit code 0
```



A screenshot of a text editor window titled 'registros.dat'. The editor displays four lines of text, each with a line number on the left. The first line is '1 NUL', the second is '2 Registro 1NUL', the third is '3 Registro 2NUL', and the fourth is '4 Registro 3'. The text is in a monospaced font, and the 'NUL' characters are underlined.

1	NUL
2	Registro 1NUL
3	Registro 2NUL
4	Registro 3

ESTRUCTURA



Preguntas de reflexión

1. ¿Qué indica el valor que devuelve `getFilePointer()` ?

El método `getFilePointer()` de la clase `RandomAccessFile` indica la posición actual del puntero dentro del archivo. Es decir, el número de bytes desde el inicio del archivo hasta la posición donde se va a leer o escribir el siguiente dato.

2. ¿Qué sucede si cambias el orden de lectura?

Cambio el orden de lectura para leer los registros desordenados obteniendo antes sus posiciones mediante `getFilePointer()`:

```
import java.io.*;

public class AccesoAleatorioOrdenLectura {
    public static void main(String[] args) {
        try {
            // Escribir registros
            RandomAccessFile raf = new
RandomAccessFile("registros.dat", "rw");
            raf.writeUTF("Registro 1");
            raf.writeUTF("Registro 2");
            raf.writeUTF("Registro 3");
            raf.seek(0);

            // Guardo en variables las posiciones de los
registros
            long pos1 = raf.getFilePointer();
            raf.readUTF(); // Lee Registro 1
            long pos2 = raf.getFilePointer();
            raf.readUTF(); // Lee Registro 2
            long pos3 = raf.getFilePointer();
            raf.readUTF(); // Lee Registro 3
            raf.seek(0); // Volver al inicio

            // Leo los registros en diferente orden:
```

```

// 1. Voy a la posición del 3er registro
raf.seek(pos3);
// Leo y muestro el 3er registro
String reg3 = raf.readUTF();
System.out.println("Registro 3: " + reg3);

// 2. Voy a la posición del 1er registro
raf.seek(pos1);
// Leo y muestro el primer registro
String reg1 = raf.readUTF();
System.out.println("Registro 1: " + reg1);

// 3. Voy a la posición del 2º registro
raf.seek(pos2);
// Leo y muestro el segundo registro
String reg2 = raf.readUTF();
System.out.println("Registro 2: " + reg2);

raf.close(); // Cierro el puntero
} catch (IOException e) {
    System.out.println("Error: " +
e.getMessage());
}
}
}

```

3. ¿Por qué `RandomAccessFile` no es recomendable para archivos de texto plano sin estructura?

`RandomAccessFile` no es recomendable para archivos de texto plano sin estructura porque estos archivos no tienen un formato fijo que permita acceder directamente a los datos que queremos, ya que las líneas pueden variar en longitud y no existen posiciones conocidas para cada registro. Esto significa que, para encontrar una línea concreta, tendríamos que leer el archivo desde el principio y contar los saltos de línea manualmente, lo cual es poco eficiente y puede generar errores si el archivo cambia. Por eso, es mejor usar clases como `BufferedReader`, que están diseñadas para leer archivos de texto línea a línea y facilitan mucho el trabajo cuando el archivo no tiene una estructura fija.

4. ¿Cómo podrías modificar solo el tercer registro sin afectar los demás?

Leo el primer y segundo registro para poder obtener la posición del 3o mediante `getFilePointer()`. Muevo el puntero a la posición del 3er registro y lo modifico con `writeUTF()`. Finalmente leo y muestro los tres registros para comprobar el resultado.

```
import java.io.*;

public class AccesoAleatorio3erRegistro {
    public static void main(String[] args) {
        try {
            // Escribir registros
            RandomAccessFile raf = new
RandomAccessFile("registros.dat", "rw");
            raf.writeUTF("Registro 1");
            raf.writeUTF("Registro 2");
            raf.writeUTF("Registro 3");

            // Calculo la posición del tercer registro
            raf.seek(0);
            raf.readUTF(); // Salta el primer registro
            raf.readUTF(); // Salta el segundo registro
            long pos3erRegistro = raf.getFilePointer();

            // Muevo el puntero a la posición del 3er registro
            raf.seek(pos3erRegistro);
            // Modifico el registro
            raf.writeUTF("MODIFICACIÓN 3er REGISTRO");

            // Compruebo el resultado
            raf.seek(0); // Muevo el puntero al inicio del
archivo
            // Leo y muestro los 3 registros
            System.out.println(raf.readUTF()); // Registro 1
            System.out.println(raf.readUTF()); // Registro 2
            System.out.println(raf.readUTF()); // Registro 3

            raf.close(); // Cierro el puntero
        } catch (IOException e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```