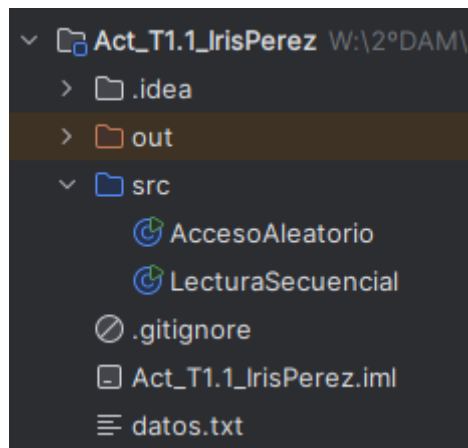


# Actividad Práctica: Tema 1.1

Iris Pérez Aparicio  
2º DAM  
Campus FP Emprende Humanes

## ESTRUCTURA DEL PROYECTO



Archivo [datos.txt](#):

```
Línea 1  
Línea 2  
Línea 3  
Línea 4  
Línea 5
```

PROGRAMA 1: [LecturaSecuencial.java](#)

```
import java.io.*;  
  
public class LecturaSecuencial {  
    public static void main(String[] args) {  
        try {  
            File archivo = new File("datos.txt");  
            BufferedReader br = new BufferedReader(new  
FileReader(archivo));
```

```

        String linea;
        System.out.println("Lectura completa del archivo
(modos secuencial):");
        while ((linea = br.readLine()) != null) {
            System.out.println("> " + linea);
        }

        br.close();
    } catch (IOException e) {
        System.out.println("Error al leer el archivo: " +
e.getMessage());
    }
}
}

```

## PROGRAMA 2: AccesoAleatorio.java

```

import java.io.*;

public class AccesoAleatorio {
    public static void main(String[] args) {
        try {
            RandomAccessFile raf = new
RandomAccessFile("datos.txt", "r");

            // Cambia el valor para probar: 0, 10, 15, 30, etc.
            long posicion = 15;
            raf.seek(posicion); // Mover el puntero al byte 15

            System.out.println("Lectura desde byte " + posicion
+ ":");
            String linea = raf.readLine(); // Leer desde esa
posición
            System.out.println("> " + linea);

            raf.close();
        } catch (IOException e) {
            System.out.println("Error en acceso aleatorio: " +
e.getMessage());
        }
    }
}

```

## PRUEBAS DE EJECUCIÓN

### PROGRAMA 1: LecturaSecuencial

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-ja
Lectura completa del archivo (modo secuencial):
> Línea 1
> Línea 2
> Línea 3
> Línea 4
> Línea 5

Process finished with exit code 0
```

### PROGRAMA 2: Acceso Aleatorio

Cambio el valor posición y compruebo donde empieza la lectura en cada caso

- seek(0)

```
Lectura desde byte 0:
> Línea 1

Process finished with exit code 0
```

- seek(10)

```
Lectura desde byte 10:
> Línea 2

Process finished with exit code 0
```

- seek(13)

```
Lectura desde byte 13:  
> nea 2  
  
Process finished with exit code 0
```

- seek(15)

```
Lectura desde byte 15:  
> a 2  
  
Process finished with exit code 0
```

- seek(20)

```
Lectura desde byte 20:  
> LÃ-nea 3  
  
Process finished with exit code 0
```

- seek(30)

```
Lectura desde byte 30:  
> LÃ-nea 4  
  
Process finished with exit code 0
```

- seek(37)

```
Lectura desde byte 37:  
> 4  
  
Process finished with exit code 0
```

- seek(45)

```
Lectura desde byte 45:  
> a 5  
  
Process finished with exit code 0
```

- seek(53)

```
Lectura desde byte 53:  
> null  
  
Process finished with exit code 0
```

### **¿Por qué cambia el resultado según el valor de seek()?**

El resultado cambia según el valor de seek porque este método mueve el puntero de lectura a la posición especificada en bytes. Ejemplo: seek(13) mueve el puntero de lectura al byte 13.