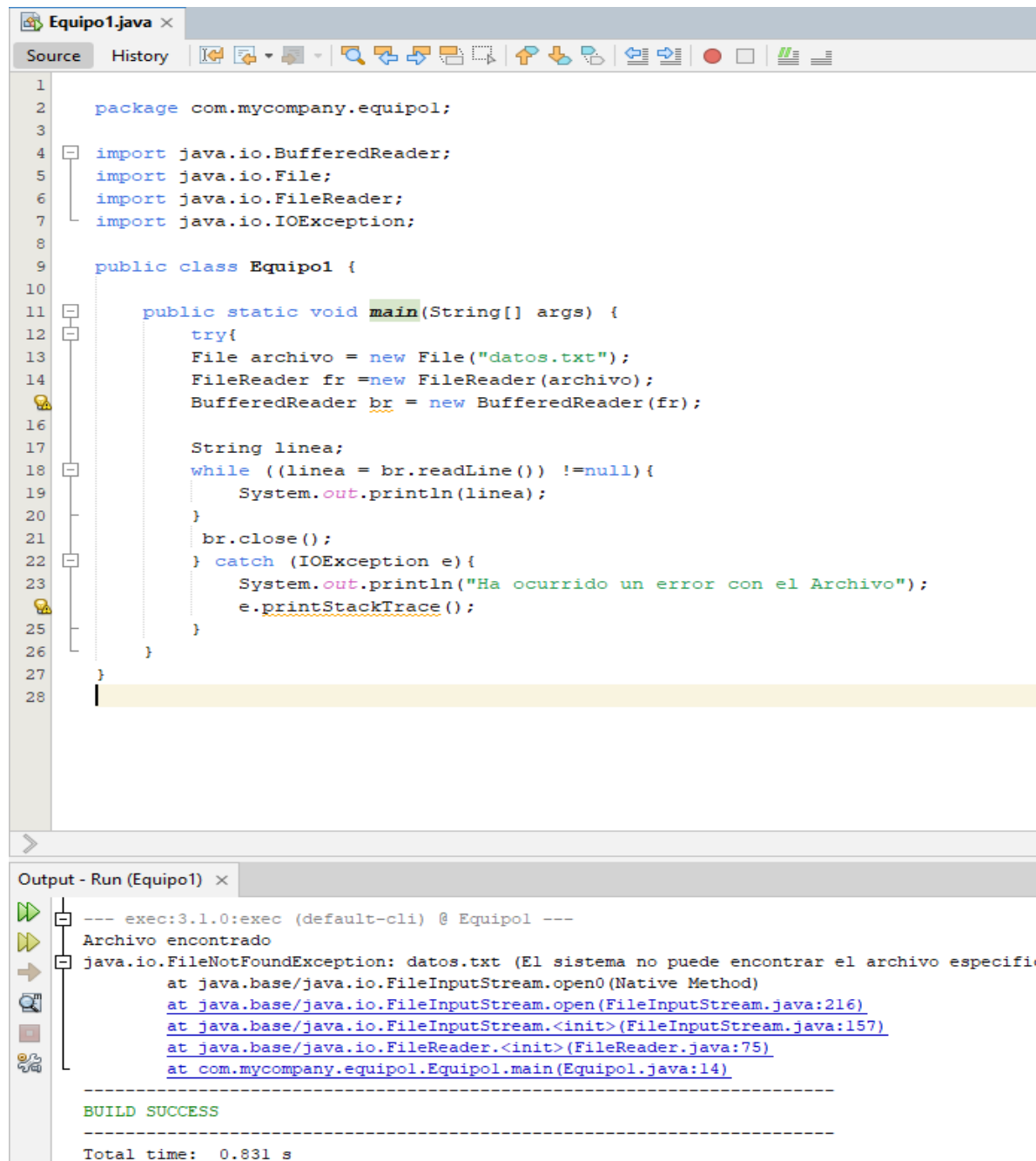


EQUIPO 1



```
1
2 package com.mycompany.equipo1;
3
4 import java.io.BufferedReader;
5 import java.io.File;
6 import java.io.FileReader;
7 import java.io.IOException;
8
9 public class Equipo1 {
10
11     public static void main(String[] args) {
12         try{
13             File archivo = new File("datos.txt");
14             FileReader fr =new FileReader(archivo);
15             BufferedReader br = new BufferedReader(fr);
16
17             String linea;
18             while ((linea = br.readLine()) !=null){
19                 System.out.println(linea);
20             }
21             br.close();
22         } catch (IOException e){
23             System.out.println("Ha ocurrido un error con el Archivo");
24             e.printStackTrace();
25         }
26     }
27 }
28
```

Output - Run (Equipo1) x

```
--- exec:3.1.0:exec (default-cli) @ Equipo1 ---
Archivo encontrado
java.io.FileNotFoundException: datos.txt (El sistema no puede encontrar el archivo especifi
at java.base/java.io.FileInputStream.open0(Native Method)
at java.base/java.io.FileInputStream.open(FileInputStream.java:216)
at java.base/java.io.FileInputStream.<init>(FileInputStream.java:157)
at java.base/java.io.FileReader.<init>(FileReader.java:75)
at com.mycompany.equipo1.Equipo1.main(Equipo1.java:14)

BUILD SUCCESS

Total time: 0.831 s
```

EQUIPO 2

```
Inventario.java X
1 import java.io.*;
2
3 public class Inventario {
4
5     static final int TAM_REGISTRO = 4 + (20 * 2) + 8 + 4; // ID + nombre (20 chars * 2 bytes) + precio + stock
6
7     public static void main(String[] args) throws IOException {
8         Scanner sc = new Scanner(System.in);
9         RandomAccessFile file = new RandomAccessFile("inventario.dat", "rw");
10
11         while (true) {
12             System.out.println("\n1. Agregar producto");
13             System.out.println("2. Consultar producto por ID");
14             System.out.println("3. Actualizar stock");
15             System.out.println("4. Salir");
16             System.out.print("Opción: ");
17             int op = sc.nextInt();
18             sc.nextLine();
19
20             if (op == 1) {
21                 System.out.print("ID: ");
22                 int id = sc.nextInt();
23                 sc.nextLine();
24                 System.out.print("Nombre (máx 20 caracteres): ");
25                 String nombre = sc.nextLine();
26                 System.out.print("Precio: ");
27                 double precio = sc.nextDouble();
28                 System.out.print("Stock: ");
29                 int stock = sc.nextInt();
30
31                 file.seek((long) id * TAM_REGISTRO); // calcula la posición exacta del registro
32                 file.writeInt(id);
33                 StringBuilder nombreFix = new StringBuilder(nombre);
34                 nombreFix.setLength(20); // asegura que el nombre tenga 20 caracteres
35                 file.writeChars(nombreFix.toString());
36                 file.writeDouble(precio);
37                 file.writeInt(stock);
38                 System.out.println("Producto guardado.");
39
40             } else if (op == 2) {
41                 System.out.print("ID del producto a consultar: ");
42                 int id = sc.nextInt();
43                 file.seek((long) id * TAM_REGISTRO);
44
45                 int pid = file.readInt();
46                 char[] nombreChars = new char[20];
47                 for (int i = 0; i < 20; i++) {
48                     nombreChars[i] = file.readChar();
49                 }
50                 String nombre = new String(nombreChars).trim();
51                 double precio = file.readDouble();
52                 int stock = file.readInt();
53
54                 System.out.println("ID: " + pid);
55                 System.out.println("Nombre: " + nombre);
56                 System.out.println("Precio: $" + precio);
57                 System.out.println("Stock: " + stock);
58
59             } else if (op == 3) {
60                 System.out.print("ID del producto a actualizar: ");
61                 int id = sc.nextInt();
62                 System.out.print("Nuevo stock: ");
63                 int nuevoStock = sc.nextInt();
64
65                 file.seek((long) id * TAM_REGISTRO + 4 + (20 * 2) + 8); // posición exacta del stock
66                 file.writeInt(nuevoStock);
67                 System.out.println("Stock actualizado.");
68
69             } else if (op == 4) {
70                 file.close();
71                 break;
72             } else {
73                 System.out.println("Opción no válida.");
74             }
75         }
76     }
77 }
78 }
```

```
Problems  @ Javadoc  Declaration  Console X
<terminated> Inventario [Java Application] C:\Users\jared\.p2\p

1. Agregar producto
2. Consultar producto por ID
3. Actualizar stock
4. Salir
Opción: 1
ID: 1231
Nombre (máx 20 caracteres): sabritas
Precio: 15
Stock: 2
Producto guardado.

1. Agregar producto
2. Consultar producto por ID
3. Actualizar stock
4. Salir
Opción: 2
ID del producto a consultar: 1231
ID: 1231
Nombre: sabritas
Precio: $15.0
Stock: 2

1. Agregar producto
2. Consultar producto por ID
3. Actualizar stock
4. Salir
Opción: 3
ID del producto a actualizar: 1231
Nuevo stock: 24
Stock actualizado.

1. Agregar producto
2. Consultar producto por ID
3. Actualizar stock
4. Salir
Opción: 2
ID del producto a consultar: 1231
ID: 1231
Nombre: sabritas
Precio: $15.0
Stock: 24

1. Agregar producto
2. Consultar producto por ID
3. Actualizar stock
4. Salir
Opción: 4
```

Equipo 3

```
MatrizOperaciones.java x Equipo2.java Equipo3.java x
1 package Sistema;
2
3 import java.io.*;
4
5 public class Equipo3 {
6     public static void main(String[] args) {
7         // Nombre a guardar
8         String nombre = "Escuela Primaria Ricardo Flores Magon";
9
10        // Nombre del archivo
11        String nombreArchivo = "datosEscuela.txt";
12
13        try {
14            // Crear FileWriter y BufferedWriter para escribir en el archivo
15            FileWriter fw = new FileWriter(nombreArchivo);
16            BufferedWriter bw = new BufferedWriter(fw);
17
18            // Escribir el nombre en el archivo
19            bw.write(nombre);
20
21            // Cerrar los recursos
22            bw.close();
23
24            System.out.println("El nombre '" + nombre + "' se ha guardado correctamente en " + nombreArchivo);
25        } catch (IOException e) {
26            System.out.println("Ocurrió un error al guardar el archivo:");
27            e.printStackTrace();
28        }
29    }
30 }
```

Javadoc Declaration Console x

<terminated> Equipo3 [Java Application] C:\Users\efren.p2\pool\plu
El nombre 'Escuela Primaria Ricardo F' se ha guardad

Equipo 4

```
MatrizOperaciones.java Equipo2.java Equipo3.java ArbolMulticaminoSimple.java x
1 package Sistema;
2
3 import java.util.ArrayList;
4
5
6
7 class NodoMulticamino {
8     List<Integer> valores;
9     List<NodoMulticamino> hijos;
10    int grado;
11
12    public NodoMulticamino(int grado) {
13        this.grado = grado;
14        this.valores = new ArrayList<>();
15        this.hijos = new ArrayList<>();
16    }
17
18    // Método para insertar un valor
19    public void insertar(int valor) {
20        // Si es nodo hoja
21        if (hijos.isEmpty()) {
22            valores.add(valor);
23            Collections.sort(valores);
24
25            // Dividir si hay más valores de los permitidos
26            if (valores.size() > grado - 1) {
27                dividir();
28            }
29        } else {
30            // Encontrar hijo adecuado
31            int i = 0;
32            while (i < valores.size() && valor > valores.get(i)) {
33                i++;
34            }
35            hijos.get(i).insertar(valor);
36        }
37    }
38 }
```

```
@ Javadoc Declaration Console x
<terminated> ArbolMulticaminoSimple [Java Applicati
Insertando valores:
23 12 44 80 17 70 25 50

Recorrido in-order del árbol:
12 17 23 25 44 50 70 80
```

Equipo 5

```
Inventario.java Registro.java Equipo5.java X
1 import java.io.IOException;
2
3
4 public class Equipo5 {
5
6     // Tamaño del registro (4 bytes para ID + 20 bytes para nombre = 24 bytes)
7     private static final int TAMANO_REGISTRO = 24;
8
9     public static void main(String[] args) {
10         String rutaArchivo = "registros.dat";
11
12         try (RandomAccessFile archivo = new RandomAccessFile(rutaArchivo, "rw")) {
13             // Escribir registros
14             escribirRegistro(archivo, 1, "Jaziel", 0);
15             escribirRegistro(archivo, 2, "Otro", 1); // Este no se usará
16
17             // Leer primer registro
18             LeerRegistro(archivo, 0);
19
20             // Actualizar registro
21             actualizarRegistro(archivo, 1, "Gonzalez", 0);
22
23             // Leer registro actualizado
24             LeerRegistro(archivo, 0);
25
26         } catch (IOException e) {
27             System.err.println("Error: " + e.getMessage());
28         }
29     }
30
31     // Método para escribir un registro
32     private static void escribirRegistro(RandomAccessFile archivo,
33                                         int id, String nombre, int posicion) throws IOException {
34         archivo.seek(posicion * TAMANO_REGISTRO);
35         archivo.writeInt(id);
36         archivo.writeBytes(formatearLongitudCadena(nombre, 20));
37     }
38
39     // Método para leer un registro
40     private static void leerRegistro(RandomAccessFile archivo, int posicion) throws IOException {
41         archivo.seek(posicion * TAMANO_REGISTRO);
42         int id = archivo.readInt();
43         byte[] nombreBytes = new byte[20];
44         archivo.readFully(nombreBytes);
45         String nombre = new String(nombreBytes).trim();
46         System.out.println("Registro: " + posicion + " - ID=" + id + ", Nombre=" + nombre);
47     }
48
49     // Método para actualizar un registro
50     private static void actualizarRegistro(RandomAccessFile archivo,
51                                         int id, String nombre, int posicion) throws IOException {
52         escribirRegistro(archivo, id, nombre, posicion);
53     }
54
55     // Método para formatear una cadena a una longitud fija
56     private static String formatearLongitudCadena(String cadena, int longitud) {
57         StringBuilder sb = new StringBuilder(cadena);
58         while (sb.length() < longitud) {
59             sb.append(' ');
60         }
61         return sb.substring(0, longitud);
62     }
63 }
```

```
Problems Javadoc Declaration Console X
<terminated> Registro [Java Application] C:\Users\jared\.p2\po
Registro: 0 - ID=1, Nombre=Jaziel
Registro: 0 - ID=1, Nombre=Gonzalez
```