**Persistència amb fitxers binaris**

**1. Idea bàsica**

* Amb fitxers **de text**, tot ho guardàvem com a text ("8.5" era caràcter 8, punt, 5).
* Amb fitxers **binaris**, Java guarda els valors en el seu format intern (per exemple, un int ocupa 4 bytes fixes).
* Això fa que sigui:
  + **Més eficient** (menys espai i més ràpid).
  + **Menys llegible per humans** (si obres el .bin amb un editor veuràs símbols estranys).

Per treballar-hi tenim dues classes clau:

* DataOutputStream → escriure dades primitives (int, double, UTF...).
* DataInputStream → llegir-les en el mateix ordre.

**Exemple bàsic: escriure i llegir valors**

* Crea una nova classe FitxerBinariExemple.java dins del teu package i enganxa:

package Paquet1;

import java.io.\*;

public class FitxerBinariExemple {

public static void main(String[] args) {

File fitxer = new File("exemple.bin");

// --- Escriure dades al fitxer binari ---

try (DataOutputStream dos = new DataOutputStream(new FileOutputStream(fitxer))) {

dos.writeInt(123); // 4 bytes

dos.writeDouble(8.75); // 8 bytes

dos.writeUTF("Hola món en binari"); // text en format UTF

System.out.println("S'han escrit dades a: " + fitxer.getAbsolutePath());

} catch (IOException e) {

e.printStackTrace();

}

// --- Llegir les mateixes dades ---

try (DataInputStream dis = new DataInputStream(new FileInputStream(fitxer))) {

int numero = dis.readInt();

double nota = dis.readDouble();

String text = dis.readUTF();

System.out.println("Llegit del fitxer:");

System.out.println(" Int: " + numero);

System.out.println(" Double: " + nota);

System.out.println(" String: " + text);

} catch (IOException e) {

e.printStackTrace();

}

}

}

Execució

1. Obre la classe a NetBeans.
2. **Run File (Shift+F6)**.
3. A l’**Output** veuràs alguna cosa així:

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

Si obres exemple.bin amb un editor de text, veuràs caràcters estranys (perquè és binari).

**Què hem après?**

* En binari, cal **llegir en el mateix ordre** que hem escrit.  
  (si escrius int + double + String, has de llegir int + double + String en aquest ordre).
* És molt útil quan vols guardar dades fixes (per exemple: id, edat, nota) sense gastar espai extra.