



Nom i Cognom:	Jonathan Rueda Neiro
Enllaç al repositori Git:	<a href="https://github.com/Zonoik09/DAM-M06-UF01-PR1.1-JRueda">https://github.com/Zonoik09/DAM-M06-UF01-PR1.1-JRueda</a>

**Objectius:**

- Aprendre a escriure i llegir arxius de text en Java mitjançant fluxos de dades, manipular fitxers en diversos formats, i aplicar bones pràctiques en la gestió i manipulació de fitxers.

**Instruccions:**

- Desenvolupa el codi necessari per a cada exercici seguint les especificacions indicades. El codi ha de superar els tests proporcionats per garantir-ne la funcionalitat.
- En cas d'haver de contestar alguna pregunta en aquest document, cal fer-ho dins del quadre indicat.

**Criteris d'avaluació:**

Cada exercici tindrà la mateixa puntuació. Es valorarà:

- Format correcte del codi (indentació i compliment de la guia d'estil de Java).
- Noms clars i descriptius per a mètodes i variables.
- Qualitat general del codi, amb comentaris explicatius quan sigui necessari.
- Les còpies seran penalitzades amb una puntuació de 0.

**Entrega:**

- Repositori Git privat, compartit amb l'usuari jpala4-ieti, que contingui el codi complet de la pràctica.
- Subdirectori "doc" dins del repositori amb el fitxer memoria.pdf.
- URL del repositori: S'ha de pujar a Moodle.

**Recursos i materials:**

- JDK de Java versió 21, Maven i Git instal·lats.
- Eina de programació: necessiteu una IDE per programar en Java (VS Code, IntelliJ, o una altra).
- Repositori amb exemples:  
<https://github.com/jpala4-ieti/DAM-M06-UF01-Repositori-Referencia-24-25> (llegiu el fitxer README.md per obtenir instruccions sobre com executar-lo).
- Altres recursos: Podeu utilitzar Google per buscar tutorials que us ajudin a resoldre els exercicis.

Punt de partida de la pràctica:

<https://github.com/jpala4-ieti/DAM-M06-UF01-PR1.1-24-25-Practica-Punt-Partida.git>



**Resol els exercicis proposats. El directori de treball ha de ser 'data', de la mateixa forma que es fa en el repositori d'exemples**

```
String camiBase = System.getProperty("user.dir") + "/data/";
```



## Exercici 0: Llegir contingut d'un fitxer i mostrar-lo per pantalla amb números de línia

Crea un programa anomenat `PR110ReadFile.java`.

Des dins del programa, realitza les següents tasques:

- Llegeix el contingut del fitxer `GestioTasques.java`, que es troba dins del directori `data`. La ruta del fitxer serà `System.getProperty("user.dir") + "/data/GestioTasques.java"`.
- Mostra per pantalla tot el contingut del fitxer, afegint el número de línia davant de cada línia de text.
- El format del número de línia serà el següent: cada línia ha d'estar precedida per un número seguit de dos punts (:) i un espai abans del contingut de la línia. Ex.: `1: Aquesta és una línia de prova.`
- Si el fitxer no existeix o es produeix un error de lectura, el programa ha de gestionar l'error i mostrar un missatge adequat, evitant que l'execució es finalitzi de manera abrupta.



## Exercici 1: Creació, manipulació i llistat d'arxius

Crea un programa anomenat `PR111Files.java`.

Des dins del programa, realitza les següents tasques:

1. Crea una carpeta anomenada `myFiles` dins del directori `data/pr111`, la qual es troba a la ruta del projecte. La ruta del directori serà `System.getProperty("user.dir") + "/data/pr111/myFiles"`.
2. Dins d'aquesta carpeta, crea dos arxius: `file1.txt` i `file2.txt`.
3. Renombra l'arxiu `file2.txt` a `renamedFile.txt`.
4. Mostra un llistat dels arxius dins de la carpeta `myFiles` amb el missatge: "Els arxius de la carpeta són:".
5. Elimina l'arxiu `file1.txt`.
6. Torna a mostrar un llistat dels arxius dins de la carpeta `myFiles` amb el missatge: "Els arxius de la carpeta són:".



## Exercici 2: Simulació de la comanda cat

Crea un programa anomenat `PR112cat.java`.

Des dins del programa, realitza les següents tasques:

- El programa haurà d'acceptar com a entrada la ruta d'un arxiu de text com a argument a l'hora d'executar-lo. La ruta del fitxer es passarà com a paràmetre en executar el programa.
- Si la ruta proporcionada correspon a un **fitxer de text**:
  - Llegeix el contingut del fitxer en **codificació UTF-8**.
  - Mostra el contingut del fitxer per pantalla.
- Si la ruta proporcionada correspon a una **carpeta**, mostra el missatge: **"El path no correspon a un arxiu, sinó a una carpeta."**
- Si la ruta proporcionada **no correspon a un fitxer existent** o es produeix qualsevol altre error, mostra el missatge: **"El fitxer no existeix o no és accessible."**

**Nota:** Gestiona adequadament possibles errors de lectura del fitxer i assegura que el programa no falli de manera abrupta. Assegura't que el contingut dels fitxers es llegeix amb la codificació UTF-8.



### Exercici 3: Creació d'un arxiu amb frases de Matrix (UTF-8)

Desenvolupa un programa que escrigui les següents frases en l'arxiu **frasesMatrix.txt**:

- "I can only show you the door"
- "You're the one that has to walk through it"

#### Requisits generals:

- El fitxer **frasesMatrix.txt** es trobarà dins el directori **data**, situat a la ruta del projecte. Utilitza **UTF-8** com a codificació per escriure i llegir l'arxiu.
- Cada frase ha de ser escrita en una nova línia, i l'arxiu ha d'acabar amb una línia en blanc.
- La ruta del fitxer es pot definir com:  
**String camiFitxer = System.getProperty("user.dir") + "/data/frasesMatrix.txt";**

#### Requisits específics:

1. **PR113sobreescriu.java:**

Aquesta versió sobreescriurà l'arxiu **frasesMatrix.txt** cada vegada que s'executi.

- Escriu les frases a l'arxiu de manera que qualsevol contingut anterior es perdi.
- Recordeu afegir una línia en blanc al final de l'arxiu.

2. **PR113append.java:**

Aquesta versió afegeix les frases al final de l'arxiu **frasesMatrix.txt** sense esborrar el contingut anterior.

- Cada execució ha d'afegir les dues frases seguides d'una línia en blanc al final del fitxer.
- Comprova que el contingut existent es manté i que es van afegint correctament les noves línies.



## Exercici 4: Generació d'un arxiu amb números aleatoris (UTF-8)

Crea un programa anomenat **PR114linies.java** que implementi aquesta funcionalitat.

- El fitxer es guardarà dins del directori **data**, situat a la ruta del projecte (**String camiFitxer = System.getProperty("user.dir") + "/data/numeros.txt";**)
- El programa ha de generar exactament **10 números aleatoris**, cada un en una línia separada.
- Utilitza la codificació **UTF-8** per escriure els números en el fitxer.
- Cada número ha de ser un valor enter aleatori generat entre 0 i 99 (ambdós inclosos).
- Assegura't que cada número estigui separat per un **salt de línia**.
- En aquest cas, després del darrer número no hi ha un salt de línia.



## Exercici 5: Simulació de la comanda cp

Crea un programa anomenat **PR115cp.java** que simuli la funcionalitat de la comanda Unix **cp** per copiar fitxers de text.

### 1. Arguments del programa:

- L'usuari haurà de passar **dues rutes** com a arguments:
  - El primer argument serà la **ruta de l'arxiu origen** que es vol copiar.
  - El segon argument serà la **ruta de destinació** on es vol guardar la còpia.

### 2. Funcionalitat:

- **Verificació de l'existència del fitxer:**
  - El programa ha de comprovar si el fitxer d'origen existeix i si és un fitxer de text.
- **Còpia del contingut:**
  - El contingut del fitxer origen es copiarà línia a línia a l'arxiu de destinació.
- **Codificació UTF-8:**
  - Totes les operacions de lectura i escriptura han de fer-se amb codificació UTF-8.
- **Línia en blanc final:**
  - Si el fitxer d'origen acaba amb una línia en blanc, s'ha de copiar tal com està, afegint també la línia en blanc al fitxer de destinació.

### 3. Advertències:

- Si l'arxiu de destinació ja existeix, el programa ha de mostrar un missatge d'advertència indicant que l'arxiu serà sobreescrit.

### 4. Comportament final:

- El programa ha de mostrar un missatge indicant que la còpia s'ha realitzat correctament, o un missatge d'error si la còpia falla.

Exemple d'ús:

```
java PR115cp /ruta/origen.txt /ruta/desti.txt
```





## Exercicis d'ampliació (opcional)

Nom del programa `PR11_ampliacio.java`.

Genera un fitxer de text amb emojis en el directori 'data' fent servir aquesta llibreria

<https://github.com/vdurmont/emoji-java>

El contingut exacte del fitxer generat el pots decidir tu mateix. S'ha d'anomenar `"emojis.txt"`