



Informàtica

**ICB0 Desenvolupament
d'aplicacions multiplataforma**

**ICC0 Desenvolupament
d'aplicacions web**

M0486 Accés a dades

AEA2 Accés a dades en fitxers

Diari d'activitats

Isidre Guixà

Curs 2025/26



Recordatori de classes usades en el mòdul M0485 (programació) a 1r. i potser alguna més	17-10-2025
<ul style="list-style-type: none"> - Per gestions en el sistema de fitxers del S.O.: <ul style="list-style-type: none"> ✓ <code>File</code>, per representar un fitxer o una carpeta. Mètodes interessants: <ul style="list-style-type: none"> <code>Constructors</code> <code>exists</code>, <code>isFile</code>, <code>isDirectory</code>, <code>createNewFile</code>, <code>delete</code> <code>getName</code>, <code>getParent</code>, <code>getAbsolutePath</code>, <code>getPath</code>, <code>lastModified</code> <code>mkdir</code>, <code>makedirs</code>, <code>listFiles</code> Camps a usar per aconseguir independència del S.O: <ul style="list-style-type: none"> <code>pathSeparator</code>, <code>separator</code>, <code>pathSeparatorChar</code>, <code>separatorChar</code> - Per gestionar les dades dels fitxers de text <ul style="list-style-type: none"> ✓ Per gestió seqüencial: <ul style="list-style-type: none"> ➢ <code>FileReader</code>, amb mètodes <code>read()</code> per llegir un caràcter o una seqüència de caràcters ➢ <code>FileWriter</code>, amb mètodes <code>write()</code> per escriure un caràcter o una seqüència de caràcters ➢ Excepcionalment <code>RandomAccessFile</code> per llegir línies via <code>readLine()</code>. ➢ <code>Scanner</code>, per llegir diferents tipus de dades (numèriques, caràcter, cadena,...) ➢ <code>PrintStream</code> i <code>PrintWriter</code>, per enregistrar diferents tipus de dades ✓ Per gestió directa per posició: <ul style="list-style-type: none"> En fitxers de text... Té sentit posicionar-nos en una determinada posició per llegir o escriure? No! De tota manera, <code>RandomAccessFile</code> facilita mètode <code>seek()</code> per posicionar-nos en una posició i llegir línia - Per gestionar fitxers binaris: <ul style="list-style-type: none"> ✓ Per escriptura seqüencial: <ul style="list-style-type: none"> <code>FileOutputStream</code>, per enregistrar bytes (cal convertir dada a enregistrar a bytes... uuff..) <code>DataOutputStream</code>, de més alt nivell doncs permet enregistrar directament dades en format binari <code>ObjectOutputStream</code>, per serialitzar objectes ✓ Per lectura seqüencial: <ul style="list-style-type: none"> <code>FileInputStream</code>, per recuperar bytes (cal convertir els bytes a dada... uuff..) <code>DataInputStream</code>, de més alt nivell doncs permet recuperar directament dades en format binari <code>ObjectInputStream</code>, per recuperar objectes serialitzats ✓ Per lectura i escriptura seqüencial i amb posicionament: <ul style="list-style-type: none"> <code>RandomAccessFile</code> <p>Per conversions entre dades i bytes (necessari si usem <code>FileOutputStream/FileInputStream</code>), donar una ullada a la classe <code>ByteBuffer</code>, que serveix de pont per passar de tipus primitius a <code>byte[]</code> i a l'inrevés.</p> <p>Mètodes de <code>ByteBuffer</code> a tenir en compte:</p> <ul style="list-style-type: none"> - <code>ByteBuffer.allocate</code> per crear un <code>ByteBuffer</code> amb una quantitat determinada de bytes. - <code>array</code> per passar el contingut d'un <code>ByteBuffer</code> a una taula de mateixa dimensió - <code>ByteBuffer.wrap</code> per crear un <code>ByteBuffer</code> a partir d'una taula - <code>putInt</code>, <code>putLong</code>, <code>put...</code> per emplenar un <code>ByteBuffer</code> amb una dada de tipus adequat - <code>getInt</code>, <code>getLong</code>, <code>get...</code> per recuperar d'un <code>ByteBuffer</code> una dada de tipus adequat. <p>Alerta! El constructor <code>FileOutputStream</code> amb només 1 paràmetre recrea el fitxer en cas d'existir. Si es vol afegir en un fitxer existent, cal usar constructor amb 2n paràmetre <code>append</code>.</p> <p>La classe <code>RandomAccessFile</code> no té la possibilitat d'indicar afegir ni tampoc recrear el fitxer, sinó que sobreescriu al damunt, de manera que si s'escriu menys informació que l'existent inicialment, el fitxer contindrà la nova informació seguida del residu de la informació original. Per això, en cas de voler crear un nou fitxer, prèviament cal comprovar l'existència del fitxer i, en cas d'existir, eliminar-lo abans d'usar <code>RandomAccessFile</code></p>	
Capes de persistència en fitxers?	17-10-2025
<p>Tenint en compte que una capa de persistència ha de permetre enregistrar, recuperar, modificar i eliminar objectes, el seu desenvolupament en fitxers hauria de ser a través d'un SGF que faciliti accés directe per valor a les seves dades i, actualment ja no s'usen, doncs han evolucionat a SGBD.</p> <p>Llavors??? Estem "obligats" a fer-ho doncs:</p> <p><i>RA1: Desenvolupa aplicacions que gestionen informació emmagatzemada en fitxers identificant-ne el camp d'aplicació i utilitzant classes específiques.</i></p> <p><i>RA6 – CA 6.c: Programa components que gestionen informació emmagatzemada en fitxers.</i></p>	



En conseqüència, desenvoluparem “*capes de persistència*”, que siguin compatibles amb les capes de persistència per a SGBD (mateixos mètodes) usant diferents tipus de fitxers (seqüencials, binaris i XML) i diferents tècniques.

Per aconseguir-ho, farem la “barbaritat” següent:

Constructors: Cercaran fitxer(s) i carregarà tota la informació en memòria (via col·leccions).

Mètodes que gestionen dades: Treballaran amb les dades existents en memòria

Mètode `commit`: Enregistrarà les dades en nous fitxers, eliminant els antics

Mètode `rollback`: Carregarà de nou totes les dades.

És un exercici purament acadèmic, amb el què practicarem la gestió de dades en fitxers.

Hem d'aconseguir que la capa a desenvolupar pugui ser usada en la darrera versió del programa de proves que hem desenvolupat en AEA1, usant doncs la mateixa interfície.

Capa `CpTxtLinEmpresa` de persistència en fitxers de text

17-10-2025

Les dades residiran en dos fitxers amb la següent estructura:

- Fitxer de text pels departaments, amb una seqüència de:
 - Línia amb el codi
 - Línia amb el nom
 - Línia amb la localitat (buida si no es coneix)
- Fitxer de text pels empleats, amb una seqüència de:
 - Línia amb el codi
 - Línia amb el cognom
 - Línia amb l'ofici (buida si no es coneix)
 - Línia amb el codi d'empleat del cap (buida si no es coneix)
 - Línia amb la data d'alta (buida si no es coneix)
 - Línia amb el salari (buida si no es coneix)
 - Línia amb la comissió (buida si no es coneix)
 - Línia amb el codi de departament (buida si no es coneix)

Es facilita fitxers `deptOriginal.txt` i `empOriginal.txt` amb les mateixes dades existents en l'esquema EMPRESA usat en AEA1 i en el format indicat en els requeriments anteriors. En executar la capa, treballarem amb duplicats `deptTreball.txt` i `empTreball.txt` d'aquests, de manera que, sempre que ens interessi, puguem partir de la situació original (com en BD on disposem de guions de creació de l'esquema i emplenat).

Els materials d'aquesta AEA2 els ubicarem a la carpeta MPO-AEA2 i aquells projectes desenvolupats en AEA1 que haguem d'usar sense evolucionar/modificar codi, els invocarem des de la carpeta MPO-AEA1 per no duplicar codi innecessari.

Veure:

- Projecte `251017_1_CpTxtLinEmpresa`, que conté els constructors, efectuant la càrrega de dades. Aquest projecte necessita els projectes:
 - `251014_2_InterCpEmpresa` de la carpeta AEA1
 - `Empresa01` de la carpeta AEA1
- Projecte `251017_1_ProvaAplicacioEmpresa`, que executa el programa `AplicacioEmpresa` usant la capa de persistència que estem desenvolupant. Aquest projecte necessita els projectes:
 - `251014_2_AplicacioEmpresa` de la carpeta AEA1
 - `251014_2_InterCpEmpresa` de la carpeta AEA1
 - `Empresa01` de la carpeta AEA1
 - `251017_1_CpTxtLinEmpresa` de la carpeta AEA2

L'execució de `ProvaAplicacioEmpresa`, en aquests moments, només efectua la càrrega de dades, doncs la resta de mètodes estan pendents de desenvolupament.

ALERTA: A classe hem suposat que el mètode `readLine` de `RandomAccessFile` retornava `null` en cas de línia buida i no és correcte. Si troba línia buida, retorna cadena buida. Només retorna `null` si no ha pogut llegir res. En el projecte `251017_1_cpTxtLinEmpresa` s'ha arreglat el codi adequadament.



Exercici: Desenvolupar la resta de mètodes de la classe CpTxtLinEmpresa	21-10-2025