Lliurament 3

Índex:

1. Objectius

En aquesta pràctica s'amplia el model de dades i s'afegeixen a l'aplicació les funcionalitats de reproducció.

Primer, haureu de definir una nova classe, ***AlbumFitxersMultimedia,*** que serà un conjunt de fitxers multimèdia amb les següents propietats:

1. Permetrà fitxers multimèdia duplicats.

2. Estarà limitat a un màxim d'N fitxers multimèdia, on N serà un valor que per defecte serà 10, però que es podrà canviar en el moment de la seva creació (es a dir, amb el mètode constructor).

3. Cada àlbum tindrà associat un títol.

L'aplicació podrà tenir una i només una biblioteca, però un nombre indeterminat d'àlbums. Els àlbums quedaran identificats pel seu títol, i per tant, no podrem tenir-ne àlbums amb el mateix títol, perquè serien àlbums repetits.

El fitxers dels àlbums han d'estar a la biblioteca; és a dir, no es poden afegir directament fitxers als àlbums si aquests no han estat afegits prèviament a la biblioteca.

A més, haureu d'afegir funcionalitats per poder reproduir-ne tant un sol fitxer com la biblioteca sencera o qualsevol dels àlbums disponibles a l'aplicació. L'usuari tindrà diferents opcions de modes de reproducció per tal de fer la reproducció d'un conjunt de fitxers de diferents maneres.

- Es definirà el mode de *reproducció continua* que, si està actiu, permetrà que quan s'acabi la reproducció (que pot ser de la biblioteca o de qualsevol àlbum) torni a començar de nou en lloc de només escoltar-la una vegada.
- També tindrem el mode de *reproducció aleatori* que, si està actiu, permetrà que els fitxers es reprodueixin en un ordre aleatori, però una sola vegada cadascun (sense repetició). Si no està actiu, els fitxers es reproduiran en l'ordre determinat a la carpeta (biblioteca o àlbum).

Per aquest lliurament, la vista del menú tindrà les següents opcions (estan en gris les opcions del menú ja implementades i en negre les que falten per implementar):

1. Gestió Biblioteca: Dona accés a un menú per a la gestió de la biblioteca.

   1.1. Afegir fitxer multimèdia a la biblioteca: Mostra un menú per tal d'afegir un fitxer multimèdia a la biblioteca.

      1.1.1. Afegir vídeo: Demana les dades necessàries per un nou vídeo.

      1.1.2. Afegir àudio: Demana les dades necessàries per un nou àudio.

      1.1.3. Menú anterior: Torna al menú anterior.

   1.2. Mostrar Biblioteca: Mostra el contingut de la biblioteca, mostrant davant de cada fitxer, el número de la seva posició a la llista començant per 1.

1.3. Eliminar fitxer multimèdia: Elimina de la biblioteca el fitxer multimèdia corresponent a una posició donada.

1.4. Menú Anterior: Torna al menú principal

2. Gestió Àlbums: Dóna accés a un menú per a la gestió dels àlbums, que conté les següents opcions:

   2.1. Afegir Àlbum: Demana les dades d'un nou àlbum i l'afegeix a l'aplicació.

   2.2. Mostar Àlbums: Mostra els àlbums disponibles a l'aplicació, mostrant davant de cada àlbum, el número de la seva posició a la llista començant per 1.

   2.3. Eliminar Àlbum: Elimina de l'aplicació l'àlbum corresponent a una posició donada.

   2.4. Gestionar Àlbum: Permet seleccionar un àlbum dels disponibles a l'aplicació, i obre un menú per a gestionar-lo. Aquest menú tindrà les següents opcions:

      2.4.1. Afegir fitxer multimèdia: Mostra els fitxers multimèdia de la biblioteca i permet seleccionar-ne un, que passarà a formar part de l'àlbum.

      2.4.2. Mostrar àlbum: Mostra el contingut de l'àlbum, mostrant davant de cada fitxer multimèdia, el número de la seva posició a la llista començant per 1.

      2.4.3. Eliminar fitxer multimèdia: Mostra els fitxers multimèdia de l'àlbum i permet seleccionar-ne un, que serà eliminat de l'àlbum.

      2.4.4. Menú Anterior: Torna al menú anterior.

   2.5. Menú Anterior: Torna al menú principal.

3. Control Reproducció/Visió: Dóna accés a un menú per al control de la reproducció o visió de fitxers, que conté les següents opcions:

   3.1. Reproduir un fitxer reproduïble: Permet reproduir un fitxer d'àudio o vídeo de la biblioteca.

   3.2. Reproduir tota la biblioteca: Permet reproduir tots els fitxers de la biblioteca.

   3.3. Reproduir un àlbum: Permet reproduir tots els fitxers d'un àlbum.

   3.4. Activar/desactivar reproducció continua: Activa i desactiva el mode de reproducció continu de la biblioteca o qualsevol àlbum.

   3.5. Activar/desactivar reproducció aleatòria: Activa i desactiva el mode de reproducció *aleatori* de la biblioteca.

   3.6. Gestió reproducció en curs: Obre un menú per la gestió de la reproducció actual, sigui de la biblioteca o d'una llista de reproducció. Aquest menú tindrà les següents opcions:

3.6.1. Re-emprèn: Re-activa la reproducció després d'una pausa o una parada.

3.6.2. Pausa: Pausa la reproducció.

3.6.3. Atura: Para la reproducció.

3.6.4. Salta: Salta el fitxer actual i passa al següent fitxer.

3.6.5. Sortir: Torna al menú anterior.

3.7. Menú Anterior: Torna al menú principal.

4. Guardar Dades: Guarda les dades de l'aplicació a un fitxer de disc.
5. Recuperar Dades: Carrega les dades de l'aplicació prèviament guardades d'un fitxer.
6. Sortir: Surt de l'aplicació.

A més, haureu de tenir en compte algunes consideracions més. Quan un fitxer multimèdia s'elimina de la biblioteca, també cal eliminar-lo dels àlbums que el contenen. Si s'elimina d'un àlbum, no s'elimina de la biblioteca.

Com en el lliurament anterior, cal que implementeu la gestió dels errors fent servir les excepcions de tipus ***AplicacioException.*** Ara aquests errors es poden produir a l'hora de les reproduccions.

2. Material pel lliurament

Per aquest lliurament us proporcionem una llibreria **UtilsProg2.jar** que conté les classes:

- ***Menu***
- ***InFileFolder***
- ***InControlador***
- ***AplicacioException***
- ***ReproductorBasic***
- ***EscoltadorReproduccioBasic***

Noteu que ara us proporcionem les interfícies ***InFileFolder*** i ***InControlador*** perquè les implementeu en les classes ***CarpetaFitxers*** i ***Controlador***, respectivament. Si aquestes classes havien estat ben definides en el lliurament 2, implementar aquestes interfícies no suposarà cap canvi en els mètodes ja implementats anteriorment.

Haureu d'utilitzar aquestes classes en el desenvolupament del lliurament. Podeu trobar **UtilsProg2.jar** al Campus Virtual i afegir-la al vostre projecte. També haureu d'afegir – igual que ho feu amb **UtilsProg2.jar** – els següents fitxers jar:

- **jna-3.52.jar**
- **platform-3.5.2.jar**
- **slf4j-api-1.7.10.jar**
- **slf4j-nop-1.7.10.jar**
- **vlcj-3.8.0.jar**

3. Descripció del lliurament

A continuació us plantegem una sèrie de passos per guiar-vos en la implementació d'aquest lliurament.

3.1. Creació del projecte (equivalent al Lliurament 1-2)

El primer pas serà crear un nou projecte, al qual li podeu posar com a nom ***Cognom1Nom1Cognom2Nom2_L3***.

Igual que en el lliurament 1, la classe principal es dirà ***IniciadorAplicacioUB***. I el paquet per defecte ***edu.ub.prog2.Congom1Nom1Cognom2Nom2.vista***. Recordeu que és possible que s'hagi d'indicar al NetBeans quina és la classe principal si no és la que s'havia establert en el moment de la creació del projecte. En aquest cas, doncs, feu que apunti a la classe ***edu.ub.prog2.Congom1Nom1Cognom2Nom2.vista.IniciadorAplicacioUB***.

3.2. Classe IniciadorAplicacioUB (equivalent al Lliurament 1-2)

La classe de la vista ***IniciadorAplicacioUB*** ha de tenir el mètode estàtic ***main*** on s'ha de crear un objecte de tipus ***AplicacioUB3*** anomenat "aplicacio" i fer la crida al mètode gestioAplicacioUB de la classe ***AplicacioUB3***:

```
aplicacio.gestioAplicacioUB ();
```

3.3. Classe AplicacioUB3 (equivalent al Lliurament 1-2)

3.3.1. Mètodes principals

El mètode d'objecte ***gestioAplicacioUB*** implementarà el menú de l'aplicació. Com en la classe del lliurament anterior aquest mètode ha de tenir la següent signatura:

```
public void gestioAplicacioUB();
```

3.3.2. Implementació del menú d'opcions i la lògica del programa

Seguint l'exemple del lliurament 1 i 2, creeu la lògica del programa, utilitzant la classe ***Menu*** de la llibreria, i tal com s'especifica en els objectius del lliurament. Comproveu especialment que podeu passar d'un menú a un altre correctament i que l'aplicació finalitza correctament.

Per modularitzar el codi de la vista i fer-lo més llegible, cal que creeu mètodes de suport en els quals es troben les funcions de cada submenú del menú principal. Recordeu també donar noms clarificadors a les constants, opcions dels *enum*s, que feu servir pels *switch* del menú.

3.4. Creació de les classes principals de l'aplicació

En aquest lliurament hem d'implementar les funcionalitats corresponents per reproduir fitxers d'àudio i vídeo. Com que la classe ***Controlador*** exerceix el control de l'aplicació, serà aquesta classe la que s'encarregarà també del control de la reproducció.

El ***Controlador*** tindrà, com fins ara, accés directe a la classe principal del model, que és ***Dades***, i un objecte de tipus ***Reproductor***. Aquest permetrà reproduir fitxers multimèdia, heretant de la classe *edu.ub.prog2.utils.**ReproductorBasic*** (el motor per reproduir fitxers).

A més, el ***Controlador*** contindrà un objecte de tipus ***EscoltadorReproduccio*** que permetrà fer reproducció de llistes de fitxers entre altres coses, i que heretarà de la classe *edu.ub.prog2.utils.**EscoltadorReproduccioBasic*** que s'ocupa de controlar els estats de la reproducció.

Com en els lliuraments anteriors la vista no manipularà directament (declarant, inicialitzant o invocant mètodes) dels objectes del model. Dit d'una altra manera: a la classe ***AplicacioUB3*** no us hauria de caldre fer-hi import de cap classe del paquet model.

3.5. Gestió dels Àlbums

A partir de les indicacions de l'enunciat, definiu la classe ***AlbumFitxersMultimedia*** que permet gestionar els fitxers multimèdia d'un àlbum. Implementeu tots els mètodes necessaris per fer la gestió dels àlbums.

Quan s'afegeix un fitxer a l'àlbum, aquest fitxer ha d'existir a la biblioteca. És a dir, l'usuari primer afegeix el fitxer a la biblioteca mitjançant el menú (opció 1.1), i després, pot seleccionar l'opció del menú per afegir fitxer a l'àlbum, i seleccionar el fitxer de la biblioteca que vol afegir.

Quan un fitxer multimèdia s'elimina de la biblioteca (opció 1.3), també cal eliminar-lo dels àlbums que el contenen. Feu servir aquí també l'excepció ***AplicacioException*** en cas que hi hagi cap problema.

Quan un fitxer s'esborra d'un àlbum, no s'esborra de la biblioteca.

3.6. Classe Dades

Com en el lliurament anterior, aquesta classe s'encarrega d'emmagatzemar les dades de l'aplicació que consistiran en una biblioteca i, ara, a més, una llista d'àlbums. Amplieu aquesta classe amb els mètodes que proveeixen les funcionalitats requerides per l'opció 2 del menú.

3.7. Reproductor

La classe Reproductor ha de definir dues versions del mètode ***reprodueix***:

```
void reprodueix(FitxerReproduible fr);

void reprodueix(Audio audio, File fitxerImatge);
```

Aquests mètodes faran ús, respectivament, dels mètodes ***play(File fitxer)*** i ***play(File fitxer, File imatge)*** de ***ReproductorBasic***. Consulteu el javadoc d'aquesta classe per més detalls de com utilitzar-la. Consulteu també el javadoc per tal de fer que el segon mètode mostri un fons negre o una imatge passada durant la reproducció d'un àudio.

Els mètodes ***reproduir*** d'***Audio*** i ***Video*** utilitzaran l'objecte de tipus **Reproductor** passat pel constructor d'aquests objectes. Quan es demani reproduir un objecte de tipus àudio o vídeo, haureu de cridar el mètode ***reprodueix*** de **Reproductor** passant-li *this*.

3.8. Classe *Controlador*

Aquesta classe vindrà guiada per la interfície que us proporcionem a la llibreria UtilsProg2, ***InControlador.*** Per això definiu aquesta classe com segueix:

```
public class  Controlador implements InControlador{

}
```

A més dels mètodes ja descrits en el lliurament anterior, hi ha nous mètodes que la interfície us obligarà a implementar. La qual cosa no vol dir que aquest siguin tots els mètodes; n'hi haurà d'altres que també haureu d'implementar i que no estaran definits a la interfície.

```
public void afegirVideo(String path, String nomVideo, String codec, float durada, int alcada, int amplada, float fps);

public void afegirAudio(String cami, String camiImatge, String nomAudio, String codec, float durada, int kbps);

public List<String> mostrarBiblioteca();  // llista dels retorns de toString() dels fitxers

public void esborrarFitxer(int id); // id és la posició a llista de getBiblioteca()

public void reproduirFitxer(int id);  // id és la posició a llista de getBiblioteca()

public void afegirAlbum(String titolAlbum);

public List<String> mostrarLlistatAlbums();

public void esborrarAlbum(String titolAlbum);

public boolean existeixAlbum(String titolAlbum);

public void afegirFitxer(String titolAlbum, int id); // id és la posició a llista de getBiblioteca()

public List<String> mostrarAlbum(String titolAlbum); // mostra informació àlbum

public void esborrarFitxer(String titolAlbum, int id); //sobrecàrrega del mètode per esborrar un fitxer d'un àlbum. id és la posició a llista de getBiblioteca()

public void guardarDadesDisc(String camiDesti);

public void carregarDadesDisc(String camiOrigen);
```

Nota: Els atributs nomVideo i nomAudio són realment les seves descripcions.

Un dels mètodes que ***InControlador*** demanarà implementar serà: ***obrirFinestraReproductor***, el qual s'implementarà amb la següent línia:

```
this.reproductor.open();
```

El mètode ***obrirFinestraReproductor*** s'ha d'invocar al començament de qualsevol mètode del controlador que hagi de realitzar tasques relacionades amb la reproducció de fitxers (opcions del menú 3.1, 3.2, 3.3 i 3.4). Així mateix, també s'haurà de cridar explícitament el mètode **tancarFinestraReproductor**; més

concretament, caldrà fer-ho just abans de tancar l'aplicació (quan es demani l'opció de sortir-ne que tenim al menú principal de l'aplicació)[1].

La reproducció s'iniciarà des d'un mètode anomenat ***iniciarReproduccio*** de la classe ***EscoltadorReproduccio*** (detallat en l'apartat 3.8 del document).

La classe ***Controlador*** haurà d'implementar també un parell de mètodes addicionals no especificats a ***InControlador*** per canviar les opcions de reproducció: activar/desactivar el mode continu i el mode *aleatori*. Feu-ne la implementació tenint en compte el següent:

- L'estat de l'opció de la reproducció (*continu* i *aleatori*) s'ha de veure reflexat quan tornem a recuperar les dades.
- Quan es fa un canvi de configuració del mode *continu* i *aleatori* aquest no tindrà efecte fins la següent vegada que es torni a iniciar una reproducció d'una carpeta de fitxers.

3.9. Classe *EscoltadorReproduccio*

Per tal de realitzar la reproducció d'un conjunt de fitxers cal crear la classe ***EscoltadorReproduccio*** que hereti de ***EscoltadorReproduccioBasic*** de la llibreria UtilsProg2. Definiu la vostra classe ***EscoltadorReproduccio*** de la següent manera:

```
public class EscoltadorReproduccio extends EscoltadorReproduccioBasic {

}
```

Aquesta classe ***EscoltadorReproduccio*** implementarà els mètodes abstractes de l'***EscoltadorReproduccioBasic*** que són cridats per ***ReproductorBasic*** quan hi ha canvis en l'estat de la reproducció. Per exemple, per aquest lliurament ens interessa el mètode ***onEndFile***, que és el que es crida automàticament quan s'acaba de reproduir un fitxer i per tant podem passar a reproduir el següent (veure apartat 3.10.).

IMPORTANT: Haureu d'utilitzar el mètode ***onEndFile*** per a fer la reproducció continuada, evitant l'ús de bucles (for/while/do-while), per què estem treballant en un model de programació orientada a esdeveniments.

[1] El botó de tancar de la pròpia finestra de reproducció no té cap efecte; això ens estalviarà possibles errors relacionats amb el maneig d'objectes gràfics de Java dels quals no volem ocupar-nos-en en aquest lliurament.

L'***EscoltadorReproduccio*** ha de tenir un mètode per iniciar la reproducció que rebrà la carpeta de fitxers a reproduir així com el mode de reproducció:

```
public void iniciarReproduccio(CarpetaFitxers llistaReproduint, boolean
reproduccioCilcica){
```

Feu la implementació de manera que pugueu re-aprofitar ***iniciarReproduccio*** tant per fitxers individuals com per reproduir la biblioteca o els àlbums. És a dir, implementeu la reproducció de fitxers individuals com carpetes d'un sol fitxer.

3.10. Reproduir fitxers de la biblioteca o d'un àlbum

La classe ***EscoltadorReproduccioBasic*** controla els diferents estats de la reproducció. El mètode ***onEndFile*** es crida automàticament sempre que s'acaba de reproduir un fitxer. Per tant, aquest mètode ens permetrà saber quan s'ha acabat la reproducció d'un fitxer i poder enllaçar la reproducció del següent.

Heu de sobreescriure el mètode ***onEndFile*** a ***EscoltadorReproduccio*** per començar la reproducció del següent fitxer cada cop que es cridi el mètode (cada vegada que acaba un fitxer). No n'heu de fer la invocació explícita, sinó que és un event intern de ***ReproductorBasic*** qui fa la invocació.

Per tal d'entendre bé com funciona el mètode ***onEndFile***, us proposem el següent exercici: sobreescriviu el mètode ***onEndFile*** a la classe **EscoltadorReproduccio** i afegiu dins d'aquest mètode una línia que imprimeixi:

```
System.out.println("S'ha acabat de reproduir el fitxer \n");
```

Després, en el mètode ***iniciarReproduccio***, que s'ha de cridar des del mètode que engega la reproducció escriviu-hi:

```
…
        ((FitxerReproduible) f).reproduir();
```

Executeu el programa i mireu que passa quan escolliu l'opció de reproduir. Feu servir el JavaDoc de la llibreria si voleu saber més sobre els mètodes del ***EscoltadorReproduccioBasic***.

3.11. Configuració dels modes de reproducció

En la classe ***EscoltadorReproduccio*** heu de controlar la reproducció, és a dir, heu de tenir en compte si la reproducció es farà amb el mode **continu** (quan s'acabi la reproducció de la llista torna a començar) o no (només s'escolta la llista una vegada) i si la reproducció es farà amb mode **aleatori** (els fitxers es reproduiran en un ordre aleatori i una sola vegada cadascuna sense repetició) o no (els fitxers es reproduiran en l'ordre determinat a la llista).

Us recomanem que abans de començar a implementar els mètodes, penseu bé l'estructura d'aquesta classe. Com ajuda us donem la llista dels atributs necessaris per facilitar el control de la reproducció en aquesta classe:

```
private CarpetaFitxers llistaReproduint;

private boolean [ ] llistaCtrl;

private boolean reproduccioCiclica, reproduccioAleatoria;
```

S'ha de controlar quins elements de la llista, que s'està reproduint, s'han reproduït ja i quins no. Per fer això, us proposem una possible implementació definint un vector de booleans del mateix tamany que la llista que s'està reproduint, inicialitzada tota a false i anar marcant cadascun dels elements (com a true) en el moment que es reprodueixen.

Aquesta implementació permetrà de forma senzilla implementar el mode aleatori de reproducció. Per generar una posició aleatòria a reproduir feu servir el mètode *random* i *round* de la classe **Math** de la següent manera:

```
pos = (int) Math.round(Math.random()*(_llistaCtrl.length-1));
```

Si aquesta posició aleatòria ja s'ha reproduït sumeu 1 a la posició.

Necessitareu implementar les funcions ***next*** i ***hasNext*** les quals us obligarà a implementar la classe pare ***EscoltadorReproduccioBasic*** de edu.ub.prog2.utils. Consulteu el JavaDoc per saber quines són les funcionalitats que han d'implementar aquestes funcions. Us recomanem que abans de començar a implementar els mètodes, penseu bé quins són els atributs necessaris per facilitar el control de la reproducció en aquesta classe.

Penseu que el mètode next ha de distingir diferents casos. Per exemple, potser s'ha arribat al final de la llista, però si tenim el mode de reproducció continu activitat, hauríem de tornar a reproduir des del principi. Si no hem arribat al final, hem de trobar el següent fitxer a reproduir.

3.12. Persistència de les dades

S'anomena *persistència* dels objectes a la seva capacitat de transcendir més enllà de l'execució actual d'un programa. Per persistir, cal que tinguin la capacitat de guardar-se i recuperar-se des d'un mitjà d'emmagatzematge.

Quan guardem objectes de tipus ***FitxerReproduible*** serà necessari utilitzar el modificador *transient* per evitar que l'atribut de tipus ***Reproductor*** es guardi juntament amb aquests objectes. Per tant, aquestes classes que declaren un atribut de tipus ***Reproductor*** ho faran de la següent manera:

```
private transient Reproductor rep;
```

El modificador *transient* permet tenir atributs no serialitzables en classes que sí ho són. Dit d'una altra manera, la classe que conté l'atribut transient podrà fer-se serialitzable, però en el moment de guardar-se/carregar-se s'ignorarà l'atribut en qüestió.

Donat doncs que el motor de reproducció no es guardarà, heu de tenir en compte que haureu de re-assignar aquest atribut als fitxers de la biblioteca quan els carregueu de disc en futures execucions de l'aplicació.

Implementeu un mètode ***setReproductor*** a la classe ***Dades*** (que es cridarà des del controlador) i que recorrerà els fitxers multimèdia per fer-los l'assignació del ***Reproductor*** a la instància actual.

3.13. Gestió d'errors

En la reproducció d'una carpeta de fitxers heu de considerar que quan un fitxer no es pot reproduir correctament, es pugui passar al següent per tal de permetre provar altres fitxers en lloc d'interrompre la reproducció amb el llançament d'una excepció.

4. Ajut pel lliurament

Per poder fer servir les classes ***ReproductorBasic*** i ***EscoltadorReproduccioBasic*** cal tenir instal·lat el VLC a la màquina i incloure al projecte de NetBeans llibreries addicionals (en format .jar) que trobareu al Campus Virtual tal i com ho feu amb **UtilsProg2.jar**.

Si voleu fer servir portàtils amb Windows, haureu de tenir instal·lada la versió experimental del VLC de 64 bits. La podeu baixar aquí:
http://get.videolan.org/vlc/2.2.2/win64/vlc-2.2.2-win64.exe

Si no utilitzéssiu la ruta d'instal·lació del VLC per defecte ("C:\\Program Files\\VideoLAN\\VLC\\"), haureu de passar-li la nova ruta al constructor del ***ReproductorBasic***.

5. Format del lliurament

El lliurament consistirà en tot el codi generat en els diferents punts de l'enunciat, juntament amb la documentació especificada en aquest apartat.

En concret, cal generar un fitxer comprimit amb el nom: **Cognom1Nom1Cognom2Nom2_L3**, que contingui:

1. El projecte sencer de NetBeans
   Tot el codi generat ha d'estar correctament comentat per a poder executar el JavaDoc, generant automàticament la documentació en línia del codi.

2. La memòria del lliurament.
   La memòria ha de contenir els punts descrits en la normativa de pràctiques i els punts següents:

1. Prepareu el diagrama de classes del vostre programa.
2. Expliqueu com heu implementat el mode continu i aleatori.
3. Expliqueu com heu implementat el mètode de la classe Controlador per reproduir un fitxer de la biblioteca i perquè.
4. Feu un diagrama de flux per mostrar el recorregut que fa el vostre programa quan s'executa l'opció d'afegir un fitxer multimèdia a la biblioteca. Especificar els mètodes que es criden en cadascuna de les classes. Feu servir fletxes i números per indicar l'ordre de les crides.
5. Expliqueu com funciona el vostre programa per a la reproducció d'una biblioteca de fitxers i fes un diagrama del flux de l'execució del programa durant aquesta reproducció.
6. Comenteu com heu utilitzat les excepcions en la gestió de la carpeta a reproduir i perquè.
7. Observacions generals.

6. Data límit del lliurament

Consulteu el calendari de pràctiques al Campus Virtual.