



PROJECTE DE PRÀCTIQUES

PROGRAMACIÓ II — LLIURAMENT 3

JOSEP NÚÑEZ RIBA

RODRIGO CABEZAS QUIRÓS

6 DE MAIG DE 2018

Índex

1. Introducció	3
2. Anàlisi.....	5
3. Desenvolupament	6
4. Qüestions	18
5. Resultats.....	20

1. Introducció

L'objectiu del projecte és desenvolupar un reproductor multimèdia, tot passant per totes les fases de desenvolupament d'un projecte de software i familiaritzar-se amb l'ús d'eines informàtiques de suport a la programació. Està enfocat cap a la programació orientada a objectes, programació orientada a esdeveniments i programació guiada per una especificació.

Els objectius a assolir en aquest lliurament són els següents:

- Implementar la classe `AlbumFitxersMultimedia`, que serà un conjunt de fitxers multimèdia amb les següents propietats:
 1. Permetrà fitxers duplicats.
 2. Tindrà una capacitat màxima d'*N* fitxers i que per defecte serà 10. Es podrà canviar amb el mètode constructor.
 3. Cada àlbum té associat el seu títol.
 4. Pot haver-hi tants àlbums com és vulguin, però només pot haver una única biblioteca.
 5. Els fitxers que es poden afegir a un àlbum han d'estar a la biblioteca.
- Definir el mode de reproducció cíclica que permet, un cop activat, que quan s'acabi la reproducció torni a reproduir des de el principi.
- Definir el mode de reproducció aleatòria que permet, un cop activat, que els fitxers es reproduueixin de forma aleatòria.
- Expandir el menú del lliurament anterior per tal de que pugui realitzar les operacions següents:
 1. Gestió àlbums (afegir, eliminar, display, gestionar àlbum i tornar al menú principal).
 2. Gestió àlbum (afegir i eliminar fitxer, display i tornar al menú anterior).

3. Control de reproducció/visió (totes les accions relacionades amb la reproducció de fitxers i tornar al menú anterior).
- Controlar les excepcions amb la classe proporcionada `AplicacioException` utilitzant els blocs de try-catch.

2. Anàlisi

Per implementar la classe `AlbumFitxersMultimedia` heretarem de la classe `CarpetaFitxers`, que ja teníem de l'anterior lliurament. Com que aquesta nova classe incorpora certes diferències respecte la classe mare, haurem de sobreesciure alguns mètodes com `isFull`, `addFitxer`, `removeFitxer` i `toString`.

Per la reproducció cíclica utilitzarem una llista de control, que s'actualitzarà cada vegada que finalitzi la reproducció d'un fitxer. D'aquesta manera sabrem quins fitxers s'han reproduït i quins no. Aquesta llista també l'utilitzarem per la reproducció aleatòria, ja que si la reproducció cíclica no és activa i el fitxer aleatori ja ha estat reproduït, haurem de triar un altre.

Actualitzar el menú simplement requerirà afegir les noves opcions requerides i ampliar el switch, es reutilitza el codi i no cal fer grans modificacions. Per les classes `Reproductor` i `EscoltadorReproduccio`, utilitzarem la llibreria `utils` proporcionada i implementarem les funcions necessàries.

Per controlar les excepcions, seguirem utilitzant el mateix sistema que hem fet servir fins ara. Abans d'implementar els mètodes de les classes, pensem en quins cassos són els que poden produir l'excepció i construïm els blocs de `throw`, `try` i `catch`.

3. Desenvolupament

- **Classe Controlador:**

- public Controlador(): Constructor de Controlador.
- public void setContinu(boolean state): Activa o desactiva la reproducció cíclica.
- public void setAleatori(boolean State): Activa o desactiva la reproducció aleatòria.
- public void setReproductor(): Setter per reproductor.
- public void afegirVideo(String path, String nomVideo, String codec, float durada, int alcada, int amplada, float fps): Afegeix un fitxer de vídeo a la biblioteca. Delega execució a Dades i a BibliotecaFitxersMultimedia.
- public Reproductor getReproductor(): Getter per reproductor.
- public void afegirAudio(String cami, String nomAudio, String codec, float durada, String camilmatge, int kbps): Afegeix un fitxer de àudio a la biblioteca. Delega execució a Dades i a BibliotecaFitxersMultimedia.
- public List<String> mostrarBiblioteca(): Mostra per pantalla un resum del contingut de la biblioteca. Delega execució a Dades i a BibliotecaFitxersMultimedia.
- public void esborrarFitxer(int id): Esborra el fitxer corresponent a la id que se li passa per paràmetre. Delega execució a Dades i a BibliotecaFitxersMultimedia.
- public void guardarDadesDisc(String camiDesti): Guarda les dades de la biblioteca al disc. Delega execució a Dades.
- public void carregarDadesDisc(String camiOrigen): Carrega les dades de la biblioteca del disc. Delega execució a Dades.
- public List<String> mostrarCamins(): Mostra els paths dels fitxers. Delega execució a CarpetaFitxers.
- public List<String> mostrarCaminsAlbum(String títol): Retorna una llista amb les rutes dels fitxers que conté un àlbum determinat.
- public boolean anyAlbums(): Comprova si hi ha àlbums.

- public boolean albumAcotat(int id): Comprova si l'id d'un àlbum existeix.
- public boolean albumAcotat(String títol, int j): Comprova donat el títol i l'id d'un àlbum si aquest existeix.
- public void esborrarAlbum(String títol): Suprimeix un àlbum determinat.
- public boolean isEmpty(): Retorna true si la biblioteca és buida. Delega execució a Dades i a CarpetaFitxers.
- public boolean isRemovable(int i): Retorna true si el fitxer és pot eliminar. I és la id del fitxer. Delega execució a Dades i a BibliotecaFitxersMultimedia.
- public void afegirAlbum(String títol, int numero): Afegeix un àlbum amb una capacitat màxima donada.
- public void afegirAlgum(String títol): Afegeix un àlbum amb la capacitat determinada.
- public List<String> mostrarLlistatAlbums(): Retorna un llistat amb tots els àlbums existents.
- public void afegirFitxer(String títol, int i): Afegeix un fitxer donat a un àlbum donat.
- public void esborrar Fitxer(String títol, int i): Elimina el fitxer del àlbum indicat.
- public boolean existeixAlbum(String titol): Comprova si existeix un àlbum determinat.
- public void reproduirFitxer(int i): Reprodueix el fitxer corresponent a la id.
- public void reproduirCarpeta(): Reprodueix tot el contingut de la carpeta.
- public void reproduirCarpeta(String títol): Reprodueix únicament l'àlbum desitjat de la carpeta.
- public void obrirFinestraReproductor(): Obre una finestra amb les comandes del reproductor.
- public void tancarFinestraReproductor(): Tanca la finestra amb les comandes del reproductor.
- public void reemprenReproduccio(): Torna a reproduir el fitxer després d'haver estat pausat.

- public void pausaReproduccio(): Pausa la reproducció del fitxer.
- public void aturaReproduccio(): Atura la reproducció.
- public void saltaReproduccio(): Reprodueix el següent fitxer disponible.
- public String id2Titol(inti d): Donada l'id d'un àlbum, retorna un string amb el seu títol.
- **Classe AlbumFitxersMultimedia:**
 - public AlbumFitxersMultimedia(String titol): Constructor, crea un àlbum amb un títol donat i capacitat per defecte (10 fitxers).
 - public AlbumFitxersMultimedia(String titol, int maxSize): Constructor, crea un àlbum amb un títol donat i capacitat desitjada.
 - public int getMaxSize(): Retorna la capacitat màxima del àlbum.
 - public String getTitol(): Getter per títol del àlbum.
 - public boolean isFull(): Retorna true si l'àlbum és ple.
 - public void addFitxer(File file): Afegeix un fitxer donat al àlbum. L'afegirà sempre i quan l'àlbum no sigui ple, i tant el directori del fitxer com aquest existeixin. En els altres cassos llençarà una excepció.

```

1.    public void addFitxer(File file) throws AplicacioException{
2.        FitxerMultimedia fitxer=(FitxerMultimedia) file;
3.        if((fitxer.exists()) && (!(fitxer.isDirectory()))){
4.            if(!(this.isFull())){
5.                this.carpeta.add(fitxer);
6.            }else{
7.                throw new AplicacioException("L'àlbum ja és
ple.");
8.            }
9.        }else{
10.            throw new AplicacioException("Aquest fitxer no
existeix.");
11.        }
12.    }

```

- public void removeFitxer(int i): Elimina un fitxer donat del àlbum.
- public void removeFitxer(File file): Elimina un fitxer donat del àlbum.
- public List<String> write(): Mostra per pantalla el títol i tot el contingut del àlbum.
- **Classe Audio:**
 - public Audio(String cami, String nom, String codec, float durada, File fitxerImatge, int kbps, Reprodutor r): Constructor de la classe Audio. Crida al constructor de la classe superior FitxerReproducible.

- public int getKbps(): Retorna els kbps del audio.
- public File getImatge(): Retorna la caràtula del audio.
- public String toString(): ToString de la classe Audio. Sobrecàrrega el mètode heretat de la classe FitxerReproducible.
- public void reproduir(): Reprodueix el fitxer.
- **Classe BibliotecaFitxersMultimedia:**
 - public BibliotecaFitxersMultimedia(): Constructor per defecte de la classe BibliotecaFitxersMultimedia.
 - public void addFitxer(File file): Afegeix un fitxer a la biblioteca. Comprova que tant el directori com el fitxer existeixen. Si el fitxer ja hi és a la biblioteca i/o no és compleixen les condicions anteriors llença una excepció.

```

1. public void addFitxer(File file) throws AplicacioException{
2.     FitxerMultimedia fitxer=(FitxerMultimedia) file;
3.     if((fitxer.exists()) && (!(fitxer.isDirectory()))){
4.         if (this.contains(fitxer)){
5.             throw new AplicacioException("Aquest fitxer ja
es troba a la biblioteca.");
6.         }else{
7.             this.carpeta.add(fitxer);
8.         }
9.     }else{
10.         throw new AplicacioException("Aquest fitxer no
existeix.");
11.     }
12. }

```

- **Classe CarpetaFitxers:**
 - public CarpetaFitxers(): Constructor per defecte. Inicialitza per defecte, la mida màxima a 100 arxius i crea un Array amb 100 slots.
 - public int getSize(): Retorna el nombre de fitxers a la carpeta.
 - public FitxerMultimedia getAt(int position): Comprova que la posició de la carpeta no és buida. En cas de no ser-ho retorna l'element corresponent en aquesta posició a la carpeta.
 - public void clear(): Buida la carpeta de fitxers.
 - public boolean isEmpty(): Si la mida de la carpeta és zero, retorna true.
 - public List<String> write(): Mostra per pantalla el contingut de la carpeta.
 - public List<String> writeCamisn(): Mostra per pantalla els directoris dels fitxers continguts en la carpeta.

- public boolean contains(FitxerMultimedia fitxer): Comprova si un fitxer és a la carpeta.
- public boolean acotat(int i): Comprova si un fitxer és a la carpeta donat el seu id.
- public void addFitxer(File fitxer): Si hi ha espai a la carpeta i l'arxiu no hi és en aquesta, afegeix el fitxer. En els altres cassos llença una excepció.
- public boolean isFull(): Comprova si la mida màxima i la utilitzada són iguals, en cas de ser-ho la carpeta és plena i retorna true. Cas contrari retorna false.
- public removeFitxer(File fitxer): Suprimeix un fitxer donat de la carpeta. Amb un bucle i un comptador, comprova que l'element iterat sigui equivalent al que es vol eliminar. Si l'arxiu és a la carpeta el borra i incrementa l'espai disponible. Cas contrari llença una excepció.

```

1. public boolean removeFitxer(FitxerMultimedia fitxer){
2.     boolean removed = false;
3.     int i =0;
4.     while((i<this.getSize())&&(!(removed))){
5.         if (fitxer.equals(this.getAt(i))){
6.             this.carpeta.remove(i);
7.             removed = true;
8.         }
9.         i++;
10.    }
11.    if(!removed){
12.        System.out.print("\n\033[31mError!
L'arxiu no existeix o no es troba en aquesta
carpeta.\033[0m\n");
13.        return false;
14.    }
15.    System.out.print("\n\033[32mFitxer eliminat
amb èxit.\033[0m\n");
16.    return true;
17.    }

```

- **Classe Dades:**

- public Dades(): Constructor per defecte de la classe Dades.
- public boolean anyAlbums(): Comprova si hi ha algun àlbum.
- public void addAlbum(String títol): Afegeix un nou àlbum.

- public void addAlbum(String títol, int i): Afegeix un nou àlbum amb capacitat màxima desitjada.
- public List<String> mostrarAlbum(String títol): Retorna una llista amb els continguts del àlbum.
- public List<String> mostrarCaminsAlbum(String títol): Retorna una llista amb les rutes dels fitxers que conté l'àlbum.
- public List<String> mostrarAlbums(): Retorna una llista amb els àlbums existents.
- public boolean isEmpty(): Retorna true si l'àlbum és buit.
- public boolean albumAcotat(int i): Donat un id, retorna true si l'àlbum corresponent existeix.
- public void esborrarAlbum(String títol): Esborra l'àlbum desitjat.
- public boolean existeixAlbum(String títol): Retorna true si l'àlbum existeix.
- public boolean albumAcotat(String títol, int i): Donat el títol i l'id d'un àlbum, retorna true si aquest existeix.
- public void afegirFitxer(String títol, int i): Afegeix el fitxer al àlbum donat.
- public void esborrarFitxer(String títol, int i): Esborra un fitxer donat de l'àlbum.
- public void esborrarFitxer(int i): Borra el fitxer donada la seva id, tant de la biblioteca com dels àlbums en que estigui inclòs.
- public List<String> mostrarBiblioteca(): Mostra per pantalla el contingut de la biblioteca.
- public boolean isEmpty(): Retorna true si la biblioteca és buida.
- public boolean isRemovable(int i): Retorna true si el fitxer corresponent a la id que se li passa pot ser suprimit.
- public List<String> mostrarCamins(): Mostra per pantalla els directoris dels fitxers que conté la biblioteca.
- public String mostrarCamins(): Mostra els paths dels fitxers que conté la biblioteca.
- public void afegirAudio(String cami, String nomAudio, String codec, float durada, String camilmatge, int kbps, Reprodutor r): Afegeix un fitxer d'àudio a la biblioteca.

```

1.     public void afegirAudio(String cami, String nomAudio, String
       codec, float durada, String camiImatge, int kbps, Reproductor
       r) throws AplicacioException {
2.         File image=new File(camiImatge);
3.         Audio
       fitxer=new Audio(cami,nomAudio,codec,durada,image,kbps,r);
4.         this.biblioteca.addFitxer(fitxer);
5.     }

```

- public void afegirVideo(String path, String nomVideo, String codec, float durada, int alcada, int amplada, float fps, Reproductor r): Afegeix un fitxer de vídeo a la biblioteca.
- public void guardar(String desti): Guarda les dades que conté la biblioteca a la direcció que se li passa per paràmetre.
- public void carregar(String origen): Carrega les dades de la biblioteca que prèviament s'han guardat. Carregarà les dades de la direcció que se li passa per paràmetre.

```

1.     public Dades
       carregar(String origen) throws AplicacioException {
2.         try{
3.             Dades model;
4.             ObjectInputStream ois;
5.             try (FileInputStream fileStream = new FileInputStrea
       m(origen)) {
6.                 ois = new ObjectInputStream(fileStream);
7.                 model=(Dades)ois.readObject();
8.                 fileStream.close();
9.             }
10.            ois.close();
11.            return model;
12.        }catch(FileNotFoundException e1){
13.            throw new AplicacioException("Fitxer no
       trobat.");
14.        }catch(IOException e2){
15.            throw new AplicacioException("Problema amb el
       tractament del fitxer.");
16.        }catch(ClassNotFoundException e3){
17.            throw new AplicacioException("Problema amb el
       casting dels objectes.");
18.        }
19.    }

```

- public void setReproductor(Reproductor r): Setter global de reproductor per a biblioteca.
- public String id2Titol(inti d): Donat l'id d'un àlbum, retorna el seu títol.
- public int titol2Id(String titol): Donat el títol d'un àlbum, retorna el seu id.

- public CarpetaFitxers reproduirFitxer(int i): Retorna una carpeta amb el fitxer a reproduir.
- public CarpetaFitxers reproduirCarpeta(): Retorna tota la biblioteca.
- public CarpetaFitxers reproduirCarpeta(String titol): Retorna l'àlbum corresponent.
- **Classe EscoltadorReproduccio:**
 - public EscoltadorReproduccio(): Constructor de la classe.
 - public void setLlista(CarpetaFitxers llistaReproduint): Setter per la carpeta de fitxers que és reproduïda.
 - public void setCiclica(boolean reproduccioCiclica): Activa o desactiva la reproducció cíclica.
 - public void setAleatorio(boolean reproduccioAleatoria): Activa o desactiva la reproducció aleatòria.
 - public CarpetaFitxers getLlista(): Getter per la carpeta que s'està reproduint.
 - public boolean[] getControl(): Retorna l'array de control.
 - public boolean getCiclica(): Retorna l'estat de la reproducció cíclica.
 - public boolean getAleatoria(): Retorna l'estat de la reproducció aleatòria.
 - public void onEndFile(): Si hi ha següent, reproduirà el següent fitxer. Si la reproducció cíclica és activa, tornarà a reproduir des de el principi. En els altres cassos s'atura la reproducció.
 - public void next(): Si la reproducció aleatòria es seleccionada, triarà una posició aleatòria que correspondrà a un fitxer de la carpeta reproduïda en l'instant. Si no és activa, augmentarà la posició una unitat per reproduir el següent fitxer a la carpeta.

```

1. public void next () {
2.     if (getAleatoria ()) {
3.         pos=(int) Math.round(Math.random()* (llistaCtrl.length
-1));
4.         while (llistaCtrl[pos]) {
5.             if ((pos+1)<llistaCtrl.length) {
6.                 pos++;
7.             } else {
8.                 pos=0;
9.             }
10.        }
11.    } else {
12.        if ((pos+1)<llistaCtrl.length) {

```

```

13.         pos++;
14.     }else{
15.         pos=0;
16.     }
17.     }
18.     FitxerReproducible
19.     fitxer=(FitxerReproducible) this.getLlista().getAt(pos);
20.     llistaCtrl[pos]=true;
21.     try{
22.         fitxer.reproduir();
23.     }catch (AplicacioException e){}

```

- public boolean hasNext(): Comprova si s'ha reproduït tota la carpeta.
- **Classe FitxerMultimedia:**
 - public FitxerMultimedia(String camí): Constructor per defecte de FitxerMultimedia. Comprova que l'arxiu indicat existeix i que es troba al path indicat (si no és compleix avisa del error). Per defecte inicialitza la descripció com a un String buit. El nom del fitxer i la seva extensió la agafa fent un substring del nom del fitxer.

```

1.     public FitxerMultimedia(String camí) {
2.         super(camí);
3.         if((this.exists()) && (!(this.isDirectory()))){
4.             this.setUltimaModificacio(new Date(this.lastMo
5. dified()));
6.             this.setCamiAbsolut(this.getAbsolutePath());
7.             String name=this.getName();
8.             int punt=name.lastIndexOf(".");
9.             this.setNomFitxer(name.substring(0,punt));
10.
11.             this.setExtensio(name.substring(punt));
12.         }else{
13.             //System.out.println("\nEl fitxer no
14. existeix.\n");
15.             this.setUltimaModificacio(new Date());
16.             this.setCamiAbsolut(camí);
17.             this.setNomFitxer("");
18.             this.setExtensio("");
19.         }
20.         this.setDescripcio("");
21.     }

```

- public FitxerMultimedia(String camí, String desc): Inicialitza tots els atributs del fitxer amb els seus valors corresponents i afegeix la descripció del fitxer.

- public void setUltimaModificacio(Data lastModified): Estableix la data d'última modificació del fitxer.
- public void setCamiAbsolut(String camí): Estableix la ruta on es troba el fitxer.
- public void setNomFitxer(String nom): Dóna nom a un fitxer donat.
- public void setExtensio(String extensió): Assigna una extensió a un fitxer.
- public void setDescripció(String desc): Dóna descripció al fitxer.
- public Date getUltimaModificacio(): Retorna la data de la última modificació.
- public String getCamiAbsolut(): Retorna la path on es troba el fitxer.
- public String getNomFitxer(): Retorna el nom del fitxer.
- public String getExtensio(): Retorna l'extensió de l'arxiu.
- public String getDescripcio(): Retorna la descripció del fitxer.
- public Boolean equals(FitxerMultimedia fitxerMultimedia): Compara els atributs d'un fitxer amb els d'un altre, si són equivalents retorna true, cas contrari retorna false.

```

1.    public boolean equals(/*Object*/FitxerMultimedia
    fitxerMultimedia){
2.        boolean return=true;
3.        if (!(fitxerMultimedia.getUltimaModificacio().equals(this
    s.getUltimaModificacio()))){
4.            return false;
5.        }else if (!(fitxerMultimedia.getCamiAbsolut().equals(this
    s.getCamiAbsolut()))){
6.            return false;
7.        }else if (!(fitxerMultimedia.getNomFitxer().equals(this.
    getNomFitxer()))){
8.            return false;
9.        }else if (!(fitxerMultimedia.getExtensio().equals(this.g
    etExtensio()))){
10.            return false;
11.        }else if (!(fitxerMultimedia.getDescripcio().equals
    (this.getDescripcio()))){
12.            return false;
13.        }else{
14.            return true;
15.        }
16.    }

```

- public String toString(): Retorna un resum amb tots els atributs del fitxer (nom, extensió, path, ...).

- **Classe FitxerReproducible:**
 - protected FitxerReproducible(String cami, String nom, String codec, float durada, Reproductor r): Constructor per la classe FitxerReproducible. Està declarat com a protected per tal de que només les subclasses d'aquesta hi puguin fer-ne ús.
 - protected abstract void reproduir(): Mètode abstracte, cal ser implementat per les subclasses.
 - public Reproductor getReproductor(): Retorna el reproductor del fitxer.
 - public String getCodec(): Retorna el codec del FitxerReproducible.
 - public float getDurada(): Retorna la durada del FitxerReproducible.
 - public String toString(): Mètode ToString de la classe FitxerReproducible. Sobrecàrrega el mètode de la classe FitxerMultimedia.
- **Classe Reproductor:**
 - public Reproductor(EscoltadorReproduccio controlador): Constructor de la classe.
 - public void reproduueix(FitxerReproducible fr): Reprodueix un fitxer donat.
 - public void reproduueix(Audio audio, File fitxerImatge): Reprodueix un àudio.
- **Classe Video:**
 - public Video(String cami, String nom, String codec, float durada, int alcada, int amplada, float fps, Reproductor r): Constructor per la classe Video. Crida al constructor de la classe superior FitxerReproducible.
 - public int getAlcada(): Retorna la alçada del fitxer de vídeo.
 - public int getAmplada(): Retorna la amplada del fitxer de vídeo.
 - public float getFps(): Retorna els *frames per second* del fitxer de vídeo.
 - public String toString(): Mètode ToString de la classe Video. Sobrecàrrega el mètode de la classe superior FitxerReproducible.
 - public void reproduir(): Reprodueix el fitxer de vídeo.
- **Classe AplicacioUB3:**
 - public AplicacioUB3(): Constructor de la classe.

- public void gestioAplicacioUB(Scanner sc): S'encarrega de gestionar les diferents opcions i accions a realitzar. Amb la classe Scanner, recull la opció seleccionada per l'usuari.
- public void gestioBiblioteca(Scanner sc): S'encarrega d'afegir, eliminar o de mostrar per pantalla els elements a la biblioteca. Amb la classe Scanner, recull la opció seleccionada per l'usuari.
- public void gestioAfegir(): Gestió de noves entrades a la biblioteca.
- public Object[] demanaVideo(Scanner sc): Demana a l'usuari les dades específiques per a afegir un fitxer de vídeo a la biblioteca.
- public Object[] demanaAudio(Scanner sc): Demana a l'usuari les dades específiques per a afegir un fitxer d'àudio a la biblioteca.
- public Object[] demana(Scanner sc): Demana a l'usuari les dades bàsiques per a afegir un fitxer a la biblioteca.
- public void gestioAlbums(): Gestions relacionades amb els àlbums.
- public void gestioContingut(String number): Gestions relacionades amb un àlbum en concret.
- public void gestioReproductor(): Gestions del reproductor.
- public void gestioCurs(): Controls del reproductor.
- public void imprimir(List<String> llista): Imprimeix llistes.
- public String pathClearing(String path): Elimina els caràcters que causen conflicte quan s'introdueixen rutes al programa.

4. Qüestions

- Explicar breument quines classes s'han pogut reutilitzar del lliurament anterior, quins canvis s'han hagut de realitzar i perquè.

Hem pogut aprofitar totes les classes tot i que hem hagut de fer alguns canvis en aquestes. Per exemple, la classe `BibliotecaFitxersMultimedia`, que pràcticament buida, ja que les seves funcions deleguen a la nova classe `AlbumFitxersMultimedia`.

- Explicar com s'ha implementat el mode de reproducció cíclica i aleatòria.

Per la reproducció aleatòria, s'agafa un valor aleatori que correspondrà al fitxer a reproduir. Amb l'ajuda de la llista de control es comprova si aquest fitxer ja ha estat reproduït, en cas afirmatiu n'agafa el següent fitxer disponible, sinó reproduceix el fitxer corresponent.

Per la reproducció cíclica hem utilitzat un array de booleans. Aquest array té la mateixa dimensió que la carpeta que s'està reproduint. Cada vegada que finalitza la reproducció d'un fitxer, s'actualitza el valor corresponent a l'array de control. Quan tots els fitxers han estat reproduïts, torna a settejar la llista i comença a reproduir de nou.

- Explicar com s'ha implementat el mètode de la classe `Controlador` per a reproduir un fitxer de la biblioteca i perquè.

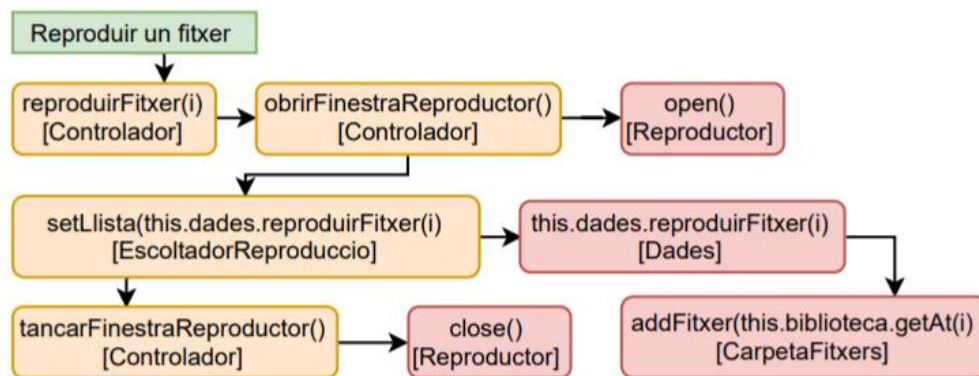
```
1.    public void reproduirFitxer(int i) throws AplicacioException
2.    {
3.        obrirFinestraReproductor();
4.        escoltador.setLlista(this.dades.reproduirFitxer(i));
5.        tancarFinestraReproductor();
6.    }
```

És rep la id d'un fitxer, s'obre la finestra del reproductor i comença la reproducció d'aquest. Un cop acabada la reproducció d'aquest es tanca la finestra. L'hem implementat d'aquesta manera per tal de que les accions

deleguin a la classe EscoltadorReproduccio, tal com indiquen les pautes del lliurament.

- Expliqueu com funciona el programa quan es sol·licita la reproducció d'una biblioteca de fitxers. Fer un diagrama de flux de l'execució del programa durant aquesta reproducció.

Un cop s'ha seleccionat el fitxer que es vol reproduir, és crida al mètode reproduirFitxer(idFitxer) que obre una finestra amb els controls del reproductor. Un cop acabada la reproducció, el mateix mètode tanca la finestra oberta prèviament i torna al menú principal de reproducció.



En taronja les crides principals durant l'execució de la reproducció, en vermell les delegacions que produeixen les crides principals en les altres classes.

- Comentar com s'han utilitzat les excepcions en la gestió de la carpeta a reproduir i perquè.

En la gestió de la carpeta que s'està reproduint hem utilitzat les excepcions per tractar el cassos crítics i que podenc causar errors en l'execució del programa. Per exemple, quan s'intenta reproduir una carpeta que és buida, és llença una excepció. Ho hem fet d'aquesta manera per tal de facilitar la cerca d'errors quan s'executa el programa, ja que quan es captura l'excepció s'imprimeix per pantalla un missatge amb l'error.

5. Resultats

- **Proves gestió dels àlbums:**

- Objectiu: Comprovar el correcte funcionament del mètode gestioAlbums.

Descripció	Resultat esperat	Resultat obtingut / Correcció
Afegir un nou àlbum.	Apareix opció d'afegir àlbum.	Correcte: Demana tots els detalls de l'àlbum i l'afegeix.
Mostrar els àlbums existents.	Mostra per pantalla els àlbums.	Correcte: Imprimeix per pantalla els àlbums amb els seus id's.
Eliminar un àlbum.	S'elimina l'àlbum.	Correcte: Àlbum eliminat.
Afegir contingut a un àlbum.	Demana dades i afegeix.	Correcte: Afegeix fitxer a la biblioteca.
Es vol veure el contingut del àlbum.	Mostra per pantalla els elements.	Correcte: Mostra per pantalla els elements.
Es vol eliminar un arxiu que està duplicat a l'àlbum.	Només un arxiu és esborrat.	Correcte: El duplicat de l'arxiu segueix a la biblioteca.
Volem tornar al menú anterior.	Surt del menú de gestió.	Correcte: Mostra per pantalla el menú principal.
Afegir un fitxer que no existeix al àlbum.	Error, el fitxer no existeix.	Correcte: Mostra per pantalla que el fitxer desitjat no existeix al disc.
Operar en un directori que no existeix.	Error, el directori no existeix.	Correcte: Explica per pantalla que el path indicat no existeix al disc.
Mostrar el contingut d'una carpeta buida.	Mostra carpeta buida.	Correcte: Mostra el missatge de que la carpeta és buida.

- **Proves gestió del reproductor:**

- Objectiu: Verificar que els controls del reproductor i la reproducció de fitxers funcionen correctament.

Descripció	Resultat esperat	Resultat obtingut / Correcció
Reproduir un fitxer.	Fitxer és reproduït.	Correcte: S'obre la finestra del reproductor i el fitxer es reproduceix.
Reproduir un àlbum buit.	L'àlbum és buit, excepció.	Correcte: Avisa per pantalla que l'àlbum és buit i no pot ser reproduït.
Activar reproducció cíclica.	S'activa la reproducció cíclica.	Correcte: Reproducció cíclica activada.
Activar reproducció aleatòria.	S'activa la reproducció aleatòria.	Correcte: Reproducció aleatòria activada.
Volem para la reproducció d'un fitxer.	La reproducció s'atura.	Correcte: El fitxer s'ha deixat de reproduir.
Final de l'àlbum amb repetició activada.	Es torna a reproduir des de el principi.	Correcte: Torna a reproduir-se l'àlbum des de el començament.
Saltem de fitxer amb la reproducció aleatòria activada en la biblioteca.	Salta a un altre fitxer qualsevol.	Correcte: Un fitxer aleatori i sense haver estat reproduït comença a reproduir-se.