



PROJECTE DE PRÀCTIQUES

PROGRAMACIÓ II — LLIURAMENT 4

JOSEP NÚÑEZ RIBA

RODRIGO CABEZAS QUIRÓS

8 DE JUNY DE 2018

Índex

1. Introducció	3
2. Anàlisi.....	4
3. Desenvolupament	5
4. Qüestions	17
5. Resultats.....	19

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.

L'objectiu a assolir en aquest lliurament és implementar una interfície gràfica amb les següents funcionalitats:

- Biblioteca: Mostrar la llista de fitxers multimèdia de la biblioteca, permetent les següents accions:
 - 1. Afegir fitxer: Permet afegir un nou fitxer a la biblioteca, ha de demanar les dades mitjançant un JDialog estàndard.
 - 2. Eliminar fitxer: Permet eliminar un o més elements seleccionats de la biblioteca.
 - 3. Reproduir:
 - a. Reproduir tota la biblioteca.
 - b. Reproduir un fitxer de la biblioteca.
- Àlbums: Mostrar la llista d'àlbums i la llista de fitxers d'un àlbum quan un és seleccionat, permetent les següents accions:
 - 1. Crear àlbum.
 - 2. Eliminar àlbum.
 - 3. Afegir fitxer a un àlbum.
 - 4. Eliminar un fitxer d'un àlbum.
 - 5. Reproduir:
 - a. Tot l'àlbum.
 - b. Un fitxer seleccionat d'aquest.
- Controls de reproducció: A la interfície cal afegir els controls necessaris per tal de definir els valors de control (reproducció continua i reproducció aleatòria), a més dels controls per tal de pausar/reprenre o parar la reproducció actual.
- Guardar i recuperar les dades.

2. Anàlisi

Per fer el menú principal, utilitzarem JButton per les diferents accions, JList per mostrar el contingut, tant de la biblioteca com dels àlbums i JComboBox per seleccionar el àlbum desitjat. Implementarem també dues funcions per actualitzar el contingut de les llistes. Això es farà cada vegada que es modifiqui la biblioteca o algun àlbum.

En el que respecta als controls de la reproducció, utilitzarem JButton per les accions que són momentànies i JToggleButton per loop i random, ja que no es desactiven automàticament.

El menú d'afegir utilitzarà JFileChooser per seleccionar els fitxers que es vol afegir i un selector que permetrà indicar quin tipus de fitxer s'afegirà. Per els fitxers d'àudio que poden tenir una caràtula, habilitarem un altre JFileChooser, aquest però només estarà habilitat quan el tipus de fitxer sigui àudio. Per facilitar la selecció de fitxers, farem que el directori default dels JFileChooser siguin les carpetes de la biblioteca de Windows "Videos", "Music" i "Pictures" respectivament. També hi posarem JTextField per indicar el path seleccionat. Donat que les dades tècniques dels fitxers no s'utilitzen, les obviarem i no les demanarem, tots els fitxers tindran els mateixos atributs específics excepte path i path imatge si té.

El menú de crear àlbums permetrà mitjançant un JTextField anomenar l'àlbum, i amb JButtons modificar la capacitat del àlbum. Utilitzarem també els toolTips de cada botó per facilitar la seva comprensió.

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 i): 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 títol): 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, Reproductor 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
fitxer=(FitxerReproducible) this.getLlista().getAt(pos);
19.     llistaCtrl[pos]=true;
20.     try{
21.         fitxer.reproduir();
22.     }catch (AplicacioException e){}
23.     }

```

- 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
dified()));
5.             this.setCamiAbsolut(this.getAbsolutePath());
6.             String name=this.getName();
7.             int punt=name.lastIndexOf(".");
8.             this.setNomFitxer(name.substring(0,punt));

9.             this.setExtensio(name.substring(punt));
10.        }else{
11.            //System.out.println("\nEl fitxer no
existeix.\n");
12.            this.setUltimaModificacio(new Date());
13.            this.setCamiAbsolut(camí);
14.            this.setNomFitxer("");
15.            this.setExtensio("");
16.        }
17.        this.setDescripcio("");
18.    }

```

- 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 AplicacioUB4:**
 - public AplicacioUB4(): Constructor de la classe.
 - protected void ompleBiblio(): Actualitzarà el contingut de la biblioteca.

- protected void ompleAlbum(): Actualitzarà el llistat dels àlbums existents.
- protected void opcionsComboBox(): Actualitzarà el JComboBox, que permet seleccionar àlbums, en funció de si s'eliminen o es creen.
- **Classe FrmGestioReproductor:**
 - public FrmGestioReproductor (AplicacioUB4 parent, Controlador controlador): Contindrà els controls del reproductor.
- **Classe FrmAfehirFitxerMultimedia:**
 - public FrmAfehirFitxerMultimedia(AplicacioUB4 parent, Controlador controlador): Contindrà el formulari per afegir nous fitxers.
- **Classe FrmNouAlbum:**
 - public FrmNouAlbum(AplicacioUB4 parent, Controlador controlador): Contindrà el formulari per a crear nous àlbums.

4. Qüestions

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

Hem reaprofitat totes les classes excepte AplicacioUB3. El motiu es simple, la versió 3 funciona amb comandes escrites per pantalla i la 4 amb GUI, per tant queda obsoleta en aquest lliurament.

- Explica quin és el model de delegació d'esdeveniments que es fa servir en el botó de reproducció de la biblioteca.

Un cop el boto ha estat premut, la delegació es la següent (→): Controlador, Dades (retorna la biblioteca) i EscoltadorReproduccio.

- Indiqueu quins tipus d'esdeveniments heu fet servir al vostre codi.
 1. JButton: botó clicat.
 2. JToggleButton: botó seleccionat (isSelected()).
 3. JComboBox: selecció de índex (getSelectedIndex()).
 4. JFileChooser: fitxer seleccionat (getSelectedFile()) i path de fitxer (getAbsolutePath()).
- Proves realitzades per comprovar el correcte funcionament de la pràctica, resultats obtinguts i accions derivades.

[Explicat al punt 5 d'aquest lliurament.](#)

- Observacions generals.

Per tal de fer que només sigui possible afegir una caràtula als àudios, hem hagut de jugar amb la funció `setEnabled`, que permet habilitar o deshabilitar elements segons convingui.

5. Resultats

- **Proves menú principal:**

- Objectiu: Comprovar el correcte funcionament del menú principal.

Descripció	Resultat esperat	Resultat obtingut / Correcció
Guardar dades.	Demana path de destí.	Error: El fitxer .dat no es crea. Correcció: Afegida línia de creació del fitxer de dades.
Carregar dades:	Demana path d'origen.	Correcte: Les dades són carregades a la perfecció.
Eliminar un fitxer d'un àlbum.	Fitxer ja no és al àlbum.	Correcte: El que hi havia menys el eliminat.
Afegir nou àlbum.	Apareix formulari de nou àlbum.	Correcte: Apareix formulari de nou àlbum.
Afegir nou fitxer.	Apareix formulari de nou fitxer.	Correcte: Apareix formulari de nou fitxer.
Reproduir biblioteca.	Inici reproducció.	Correcte: Inici de la reproducció de la biblioteca.
Reproduir fitxer.	Reprodueix el fitxer.	Error: No es reprodueix. Correcció: La biblioteca era buida, hem afegit un nou fitxer.
Controls de reproducció.	Apareixen els controls.	Correcte: Els controls apareixen.
Eliminar àlbum que no existeix.	No fa res.	Correcte: No hi ha cap àlbum per esborrar.
Reproduir àlbum.	Es reprodueix	Correcte: S'inicia la reproducció del àlbum seleccionat.

- **Proves controls de reproducció:**

- 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.

- **Proves menú nou àlbum:**

- Objectiu: Verificar el correcte funcionament del formulari de creació d'àlbums.

Descripció	Resultat esperat	Resultat obtingut / Correcció
Crea un àlbum amb 0 capacitat.	El mínim és 1.	Correcte: Només deixa fer àlbums amb 1 de capacitat.
Crea un àlbum sense nom.	No deixa.	Correcte: No l'afegeix fins que se li indica un nom.
Cancel·lar la creació del àlbum.	Finestra es tanca.	Correcte: L'àlbum no s'afegeix i tornem al menú principal.

- **Proves menú afegir nou fitxer:**
 - Objectiu: Verificar el correcte funcionament del formulari de creació d'àlbums.

Descripció	Resultat esperat	Resultat obtingut / Correcció
Afegir un vídeo.	Afegeix vídeo.	Correcte: Habilita selector path del vídeo i l'afegeix.
Afegir un àudio.	Afegeix àudio.	Correcte: Habilita selector path del àudio i de la imatge i l'afegeix.
Afegir un àudio sense caràtula.	L'afegeix.	Correcte: El paràmetre de la imatge no és obligatori.
Afegir un vídeo sense seleccionar path.	Error.	Correcte: Avisa a l'usuari de que no ha seleccionat cap fitxer.
Cancel·lar operació.	Torna al menú principal.	Correcte: No afegeix cap fitxer i retorna tanca la finestra.