



UNIVERSITAT DE  
BARCELONA

## *Lliurament 3*



**UNIVERSITAT DE BARCELONA**

**LABORATORI PROGRAMACIÓ II**

**Lliurament 3**

**Grup B00**

**Jeroni Carandell**

Pau Bernabé **NIUB:** 20081736

Nil Viñals **NIUB:** 20100522

17/05/2018

1. Expliqueu quines classes has pogut reutilitzar del lliurament anterior per a fer aquest. Quins canvis sobre les classes reutilitzades heu necessitat fer i perquè.

**Del paquet controlador:** Hem reutilitzat la majoria de mètodes de la classe Controlador i n'hem afegit com ens indicava InControlador.

De la classe Reproductor hem reutilitzat també la classe i afegit algun mètode més.

**Del paquet model:** Hem reutilitzat tot, bàsicament només hem implementat més mètodes a Dades i hem creat la classe Àlbums.

**Del paquet vista:** Hem reutilitzat tot, a AplicacioUB3 només hem afegit uns quants mètodes.

2. Expliqueu com heu implementat el mode continu i aleatori.

El mode continu de reproducció l'hem implementat de manera que a cada reproducció de la carpeta, anés marcant la variable índex de la llistaCtrl com a true, indicant que ja s'havia reproduït. Quan arribàvem a tenir la carpeta tota en true, llavors tornàvem a omplenar-la a false i posar la variable índex a 0. Com es volia en aquest lliurament, el continu havia de donar la volta a la llista de reproducció i així és.

3. Expliqueu com heu implementat el mètode de la classe Controlador per reproduir un fitxer de la biblioteca i perquè.

El mètode per reproduir un sol fitxer de la biblioteca l'hem creat de manera, que compleixi amb els paràmetres del mètode iniciarReproduccio. En aquest mètode esmentat anteriorment li passàvem una carpeta de fitxers, així que per a cada fitxer individual, crearem una nova carpeta que el contingui i es reproduceixi.

```
public void reproduirFitxer(int i) {  
    CarpetaFitxers unic=new CarpetaFitxers();  
    try{
```

```
unic.addFitxer(dades.getBiblio().getAt(i));  
this.obrirFinestraReproductor();  
this.er.iniciarReproduccio(unic, dades.getContinua(), dades.getRandom());  
}catch(AplicacioException e){  
    e.getMessage();  
}
```

El mode de reproducció aleatòria funciona de manera que si el metode hasNext() i l'atribut de reproduccioAleatoria estan a true, primer comprovarem si la posició aleatoria que hem generat amb l'atribut pos es falsa, si no ho és, farem una altra vegada next() fins que trobi una posició reproduïble. Llavors quan acabem i no trobem més posicions reproduïbles, aquest deixarà sonant l'última cançó. No fa el loop fins a la primera posició.

```
}else if(hasNext() && reproduccioAleatoria){  
    File fitxer=llistaReproduint.getAt(pos);  
    if(llistaCtrl[pos]!=false){  
        next();  
    }else{  
        ((FitxerReproduible)fitxer).reproduir();  
        llistaCtrl[pos]=true;  
    }  
}
```

#### 4. 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.

Suposant que ja s'ha inclòs fitxers i creat àlbums, vas a Control Reproducció, una vegada estàs allà, tens l'opció "Reproduir un fitxer reproduïble", "Reproduir tota la biblioteca", "Reproduir un àlbum", "Activar/desactivar reproducció continua", "Activar/desactivar reproducció aleatòria", "Gestió reproducció en curs", en aquest últim s'obrirà un submenú respecte el control de la reproducció amb: "Re-emprèn", "Pausa", "Atura" i "Salta".

**Reproduir un fitxer reproduïble:** Aquest mètode s'imprimirà la biblioteca perquè puguem veure els fitxers que hi ha dins i a continuació obtenint la id del fitxer li passarem el mètode controlador.reproduirFitxer(id-1) amb l'objecte controlador i decremantant 1, ja que l'array de la llista comença en 0 i en imprimir per 1, agafarem i crearem una carpetaFitxers amb un sol element, obrirem una finestra per la reproducció del file i passarem la llista pel parametre de iniciarReproduccio(CarpetaFitxers llistaReproduint, boolean continua, boolean random), que es troba a l'escoltador, aquest anirà reproduint l'àudio i es parará quan acabi, és a dir que farà un next però al no haver següent es parará.



**Reproduir tota la biblioteca:** Aquest mètode, crida directament a un mètode dins de controlador que fa l'iniciarReproduccio, com hem fet anteriorment passant com a parametre la biblioteca de dades, qué es on hem guardat tots els fitxers fent així que travessi a dins d'escollidorReproduccio tot l'array i al tenir next cada file res reproduceix fins al final d'aquest.

**Reproduir un àlbum:** Aquest mètode, et pregunta el títol per passar per paràmetre a reproduirCarpeta a través d'un objecte controlador, al coincidir el nom fa una recerca per tots els àlbums fins que trobi un amb el mateix nom i el passarà a través de iniciarReproductor com a paràmetre com a l'anterior mètode.

**Activar/desactivar reproducció continua:** Fa que l'atribut reproducció continua s'inverteixi cada vegada que s'accedeix aquest mètode.

**Activar/desactivar reproducció aleatòria:** Fa que l'atribut reproducció aleatòria s'inverteixi cada vegada que s'accedeix aquest mètode.

**"Re-emprèn", "Pausa", "Atura" i "Salta":** Aquests mètodes tindran crides individuals a mètodes dins de controlador que utilitzen mètodes fets ja per la classe ja donada ReproductorBasic.

*(Diagrama de flux dins del .zip, en un fitxer apart)*

## 5. Comenteu com heu utilitzat les excepcions en la gestió de la carpeta a reproduir i perquè.

Hem fet una modificació per les Excepcions dins de iniciarReproduccio per si vols reproduir una carpeta buida, et surti el missatge, ja que no es possible reproduir una cosa que no sigui un reproducible i menys un element buit.

## 6. Observacions generals:

En aquest lliurament hem trobat alguna dificultat en solventar certs problemes de NullPointerException, coses mal inicialitzades feien que certs mètodes funcionessin malament. Finalment i amb alguns inconvenients que ens han sorgit als dos quan estàvem en grups diferents, hem pogut tirar endavant la pràctica 3.