UNIVERSITAT DE BARCELONA

FACULTAT DE MATEMÀTIQUES I INFORMÀTICA

MEMÒRIA DEL QUART LLIURAMENT DE PRÀCTIQUES

PROGRAMACIÓ II

Martí Pedemonte i Bernat

28 de maig de 2017

Índex

Introducció

El principal objectiu d'aquest quart lliurament de pràctiques és familiaritzar-se amb el disseny d'interfícies gràfiques. Per això en aquesta entrega utilitzarem la programació orientada a esdeveniments, que ens permetrà generar una interfície per executar el programa del lliurament anterior utilitzant controls gràfics i no des de la consola del NetBeans. Seguint el guió que se'ns ha proporcionat, haurem d'implementar tres finestres principals:

- **Finestra principal de la sessió d'un usuari:** Aquesta finestra serà el menú principal de l'usuari. S'hi podrà visualitzar tant el repositori privat com el públic, i tindrà els següents controls:
  - Botons per afegir un nou fitxer al repositori privat i suprimir, compartir i reproduir un fitxer seleccionat del repositori privat.
  - Botons per controlar la reproducció en curs (reprèn, pausa, atura i salta, així com un botó seleccionable per controlar la reproducció cíclica).
  - Botons per reproduir els repositoris, un pel privat i un pel públic.
  - Botons per guardar i recuperar dades.
  - Un botó per canviar d'usuari.
- **Finestra d'entrada d'usuari:** Aquesta finestra s'obrirà només iniciar el programa, i demanarà les dades per a registrar un nou usuari o bé per entrar un usuari ja existent. També s'obrirà al prémer el botó de canviar d'usuari del menú principal.
- **Finestra d'addició d'un nou fitxer multimèdia:** Aquesta finestra s'obrirà al prémer el botó per afegir un nou fitxer al repositori privat de l'usuari, i obrirà un formulari per entrar les dades del fitxer a afegir.

La tasca important serà anar cridant els diversos mètodes del controlador (que recuperarem del lliurament anterior, de la mateixa manera que el model) des dels controls de la interfície gràfica per poder dur a terme totes les accions anteriorment esmentades.

Capítol 1

Anàlisi i desenvolupament

1.1 Anàlisi

Per assolir els objectius prèviament descrits, hauré de dissenyar el programa abans d'escriure cap línia de codi. He de definir cada classe i el seu funcionament, sense entrar en detalls tècnics, ja que això ho explicaré més endavant, a la secció de desenvolupament.

Així doncs, explicaré cada classe nova i cada modificació de les classes dissenyades en anteriors lliuraments, si es que n'hi ha.

En primer lloc, definiré tres classes més, totes a la vista. Aquest lliurament només tractarà d'això, ja que totes les funcionalitats del programa ja van ser dissenyades i implementades al lliurament anterior. És a dir, els paquets model i controlador es recuperaran i no es tocaran per a res, ja que només canviarà la manera com l'usuari interacciona amb l'aplicació. Les tres noves classes (que són les úniques que constaran a la vista) són **AppUB4**, **FrmRegistreInici** i **FrmAddMediaFile**.
AppUB4 serà el menú principal de l'usuari, **FrmRegistreInici** serà la finestra d'entrada d'usuari i **FrmAddMediaFile** serà la finestra per afegir fitxers al repositori privat de l'usuari. Així doncs, passo a definir-les en detall.

1.2 Desenvolupament

En aquesta secció explicaré més detalladament els mètodes i atributs de les classes creades per dur a terme aquesta entrega.

1.2.1 Classe **FrmRegistreInici**

Aquesta nova classe ens permetrà registrar nous usuaris i entrar-ne un de ja registrat. Com he esmentat anteriorment, estarà al paquet de la vista. Aquesta classe estendrà la classe **JDialog**, classe per a dissenyar interfícies gràfiques de Java. Contindrà un objecte de tipus **Controller** que se li passarà per paràmetre al constructor, i que, com veurem més endavant, serà el mateix objecte per a les tres

noves classes. A més a més contindrà dos botons (**JButton**), un per a registrar i l'altre per entrar un usuari, i dues etiquetes (**JLabel**) per a enumerar els dos camps de text (**JTextField**) on l'usuari entrarà el seu nom i el seu ID.
Els mètodes que dependran d'un esdeveniment seran els següents:

- btnSignInActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Sign in". Comprovarà que no existeixi l'usuari entrat, i si és així, registrarà aquest usuari i obrirà un **JOptionPane** informant de que s'ha registrat correctament. En el cas que l'usuari ja existís, obrirà un **JOptionPane** amb un missatge d'error.

- btnLogInActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Log in". Comprovarà si existeix l'usuari entrat, i si és així, assignarà aquest usuari com a actual i tancarà la finestra. En el cas que no existís, obrirà un **JOptionPane** amb un missatge d'error.

Aquesta finestra de registre d'usuari no es podrà tancar si no entra un usuari, ja que no tindria sentit accedir al menú d'usuari sense haver-ne entrat un. L'aspecte gràfic que tindrà aquesta finestra serà el següent:

Figura 1.1: Aspecte gràfic que tindrà aquesta finestra.

1.2.2 Classe FrmAddMediaFile

Aquesta nova classe ens permetrà afegir fitxers multimèdia al repositori privat de l'usuari. Com a l'anterior classe, també estendrà **JDialog**, i contindrà un objecte de tipus **Controller** que se li passarà per paràmetre al constructor. A més a més contindrà quatre botons (**JButton**), un "combo box" (**JComboBox**), deu etiquetes (**JLabel**) i nou camps de text (**JTextField**).
En primer lloc, el "combo box" permetrà seleccionar si el fitxer que es vol afegir és de vídeo, àudio o bé una imatge. Aleshores, depenent l'opció escollida, s'activaran els camps necessaris per a afegir el fitxer. Cada camp de text anirà al costat de la seva etiqueta corresponent, i en el cas de la ruta del fitxer i la ruta de la fotografia del fitxer d'àudio també hi haurà un botó que permetrà seleccionar el fitxer en qüestió. Finalment, hi haurà un botó d'acceptar (que només estarà actiu si s'han emplenat tots els camps necessaris) i un botó de cancel·lar.
Els mètodes que dependran d'un esdeveniment seran els següents:

- btnCancelActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Cancel". Simplement tancarà la finestra.

- btnAcceptActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Accept". Afegirà el fitxer del qual s'han donat les dades al repositori privat de l'usuari actual i tancarà la finestra. En cas que en un camp on només s'hi poden entrar números s'hi entrin altres caràcters, mostrarà un **JOptionPane** amb un missatge d'error, i no afegirà el fitxer ni tancarà la finestra.

- btnOpenFileActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Open file". Obrirà una finestra de diàleg de tipus **JFileChooser** on l'usuari escollirà un fitxer, i un cop escollit s'anotarà la ruta al camp de text adjacent al botó.

- btnOpenThumbnailActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Open file" (del camp etiquetat com a "Thumbnail"). Obrirà una finestra de diàleg de tipus **JFileChooser** on l'usuari escollirà una imatge, i un cop escollida s'anotarà la seva ruta al camp de text adjacent al botó.

- cmbFileTypeItemStateChanged: Aquest mètode s'accionarà quan es detecti algun canvi al "combo box". Activarà els camps corresponents a l'opció seleccionada d'aquest control, és a dir: si l'opció escollida és "Video", s'activaran els camps de text corresponents al codec, duració, alçada, amplada i fotogrames per segon. Si l'opció escollida és "Audio", s'activaran els camps de text corresponents al codec, duració, caràtula (i el seu botó) i kilobits per segon. Si l'opció escollida és "Image", s'activaran els camps de text corresponents a l'alçada i l'amplada. Si és la primera vegada que es selecciona alguna opció diferent de la inicial ("null"), també s'activaran els camps corresponents a la ruta del fitxer (i el seu botó) i la seva descripció.

- Els mètodes següents: txtFilePathCaretUpdate, txtDescriptionCaretUpdate, txtCodecCaretUpdate, txtDurationCaretUpdate, txtHeightCaretUpdate, txtWidthCaretUpdate, txtFpsCaretUpdate, txtThumbnailCaretUpdate, txtKbpsCaretUpdate. Aquests mètodes s'accionaran cada vegada que detectin un canvi en el seu camp. Quan el detectin, totes cridaran el mètode enableAcceptButton, que ara descriuré.

Els mètodes que no dependran de cap esdeveniment seran els següents:

- omplirLlista: Mètode únicament cridat al constructor de la classe. Omple la llista del "combo box" per assignar tres possibles valors: "Video", "Audio" i "Image".

- enableAcceptButton: Aquest mètode és cridat cada vegada que algun camp rep un canvi d'estat. Comprovarà si tots els camps són plens depenent del tipus de fitxer seleccionat al "combo box", i en cas afirmatiu activarà el botó. Si hi ha algun camp per omplir, no s'activarà. D'aquesta manera s'obliga a l'usuari a omplir totes les dades del fitxer abans d'afegir-lo.

L'aspecte gràfic que tindrà aquesta finestra serà el següent:

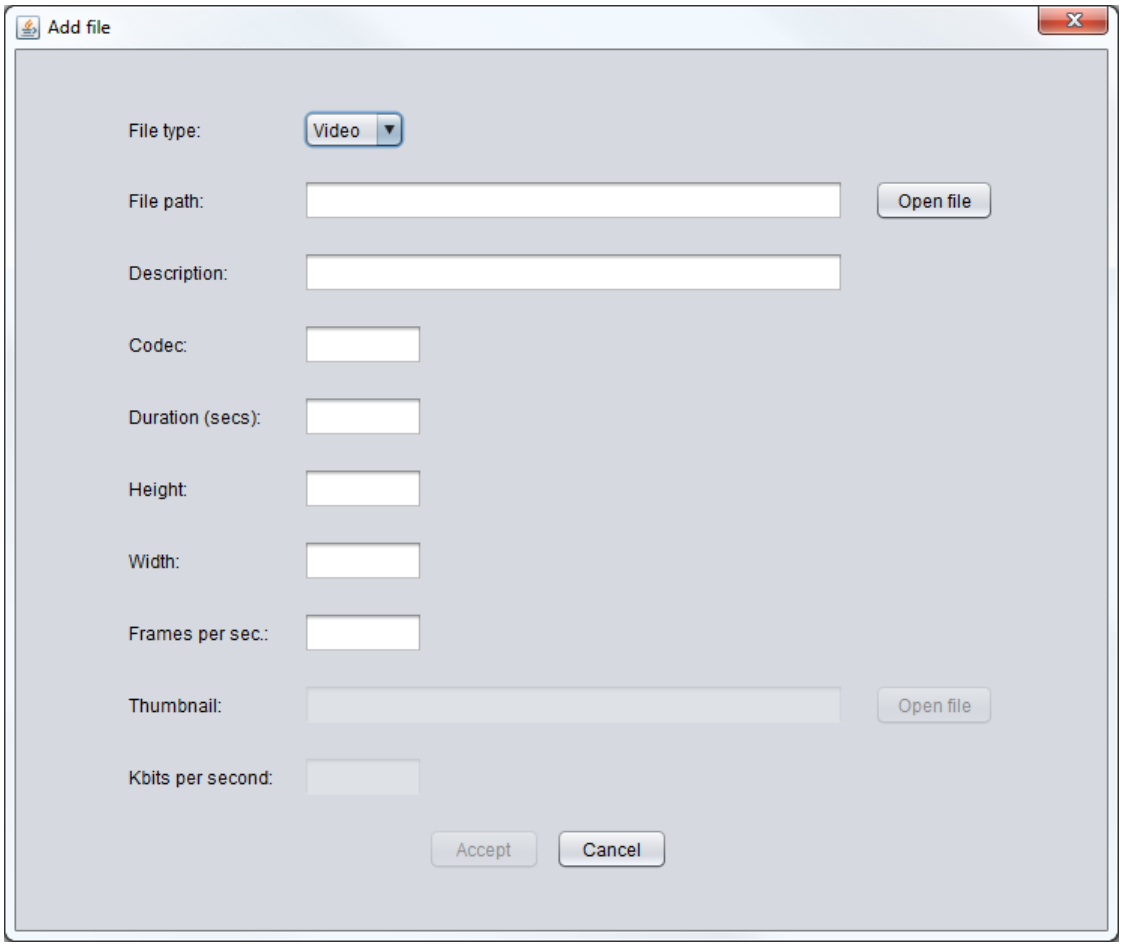

Figura 1.2: Aspecte gràfic que tindrà aquesta finestra.

1.2.3 Classe **AppUB4**

Aquesta nova classe ens permetrà gestionar el menú principal de l'usuari. A diferència de les altres dues classes implementades en aquesta entrega, aquesta estendrà **JFrame**, que també és una classe per a dissenyar interfícies gràfiques de Java. Contindrà un objecte de tipus **Controller** com a atribut principal, que serà el que se li passarà com a paràmetre per constructor a les altres dues classes, ja que aquesta serà la classe base perquè contindrà el mètode main. A més a més contindrà tretze botons (**JButton**), un botó seleccionable (**JToggleButton**), dues etiquetes (**JLabel**) i dues llistes (**JList**) (recordem que per a poder existir un objecte d'aquest tipus, és necessària la creació d'un objecte tipus **JScrollPane** que el contingui). El constructor inicialitzarà tots els components i l'objecte **Controller**.
Els botons principals, a l'esquerra de la finestra, seran els d'afegir un nou fitxer al repositori privat, eliminar i reproduir un fitxer del repositori privat i compartir un fitxer del repositori privat al públic. Aleshores, a la dreta d'aquests controls hi haurà els cinc botons que gestionaran la reproducció: reprèn, pausa, atura, salta i activa/desactiva la reproducció cíclica (aquest últim serà el botó seleccionable). Més a la dreta i una mica amunt, hi trobarem els botons de canvi d'usuari, i de guardar i recuperar

les dades.
Sota de tots aquests botons s'hi podrà veure dues etiquetes, mostrant repositori privat i repositori públic, respectivament, i per sota d'aquestes s'hi veuran les dues llistes, una per repositori, mostrant en temps real els fitxers existents. A més a més, sota de cada llista hi haurà el botó de reproducció completa del repositori. Els mètodes que dependran d'un esdeveniment seran els següents:

- btnChangeUserActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Change user". Obrirà la finestra de registre d'usuari, cridant el mètode openRegWindow que explicaré més endavant.

- lstPrivRepValueChanged: Aquest mètode s'accionarà cada vegada que es detecti algun canvi en la selecció d'elements del repositori privat. Comprovarà que la selecció no és nul·la, i de ser així, permetrà l'activació dels botons "Delete file", "Share file" i "Play file".

- btnAddFileActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Add file". Obrirà la finestra d'addició de fitxers, cridant el mètode openAddFileWindow que explicaré més endavant.

- btnDeleteFileActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Delete file", que només estarà actiu quan hi hagi un o més fitxers seleccionats al repositori privat. Obrirà una finestra de diàleg de tipus **JOptionPane** on l'usuari confirmarà que vol eliminar els fitxers seleccionats del repositori privat, i seguidament els eliminarà. Després cridarà els mètodes pubRepFill i privRepFill, que ompliran de nou les llistes, i que també veurem en detall a posteriori.

- btnShareFileActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Share file", que només estarà actiu quan hi hagi un o més fitxers seleccionats al repositori privat. Compartirà aquests fitxers i cridarà els mètodes pubRepFill i privRepFill.

- btnPlayFileActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Play file", que només estarà actiu quan hi hagi un o més fitxers seleccionats al repositori privat. Si hi ha més d'un fitxer seleccionat, obrirà una finestra de diàleg de tipus **JOptionPane** on s'avisarà a l'usuari que no es poden reproduir dos fitxers a la vegada, i que només en pot seleccionar un. En cas que només n'hi hagi un de seleccionat, el reproduirà i cridarà el mètode refreshPlayButtons que explicaré més endavant.

- tbtnCiclicStateChanged: Aquest mètode s'accionarà quan es detecti algun canvi en el botó seleccionable de reproducció cíclica. En cas que estigui seleccionat, modificarà el booleà del controlador corresponent a la reproducció cíclica com a verdader, i si està desactivat, com a fals.

- btnPlayPrivRepActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Play private repository". Si hi ha algun fitxer al repositori, començarà la reproducció d'aquest i cridarà el mètode refreshPlayButtons. Si no n'hi ha cap, obrirà un quadre de diàleg de tipus **JOptionPane** anunciant l'error.

- btnPlayPubRepActionPerformed: Aquest mètode s'accionarà quan es premi el botó de "Play public repository". Si hi ha algun fitxer al repositori, començarà la reproducció d'aquest i cridarà el

mètode refreshPlayButtons. Si no n'hi ha cap, obrirà un quadre de diàleg de tipus **JOptionPane** anunciant l'error.

- btnResumeActionPerformed: Aquest mètode s'accionarà quan es premi el botó de reprendre la reproducció del fitxer pausat o aturat, i el rependrà. Cridarà el mètode refreshPlayButtons.
- btnPauseActionPerformed: Aquest mètode s'accionarà quan es premi el botó de pausar la reproducció del fitxer en curs, i el pausarà. Cridarà el mètode refreshPlayButtons.
- btnStopActionPerformed: Aquest mètode s'accionarà quan es premi el botó d'aturar la reproducció del fitxer en curs, i l'aturarà. Cridarà el mètode refreshPlayButtons.
- btnSaveDtaActionPerformed: Aquest mètode s'accionarà quan es premi el botó "Save data". Obrirà un quadre de diàleg de tipus **JFileChooser** i l'usuari triarà on vol guardar les dades.
- btnLoadDataActionPerformed: Aquest mètode s'accionarà quan es premi el botó "Load data". Obrirà un quadre de diàleg de tipus **JFileChooser** i l'usuari triarà des d'on vol recuperar les dades. Un cop recuperades, cridarà el mètode openRegWindow perquè pugui entrar un usuari.
- windowOpened: Aquest mètode s'accionarà quan s'obri la finestra de l'aplicació per primera vegada. Només cridarà el mètode openRegWindow perquè pugui entrar un usuari. Noteu que és un mètode sobreescrit de la interfície **WindowListener**.
- windowClosed: Aquest mètode s'accionarà quan es tanqui la finestra de l'aplicació. Si hi ha una finestra de reproducció oberta, la tancarà i sortirà. Si no n'hi ha cap d'oberta, simplement sortirà. També és un mètode sobreescrit de la interfície **WindowListener**.
- formMouseMoved: Aquest mètode s'accionarà cada vegada que es mogui el ratolí. Simplement cridarà el mètode refreshPlayButtons per actualitzar l'estat dels controls de reproducció.

Els mètodes que no dependran de cap esdeveniment seran els següents:

- refreshPlayButtons: Mètode creat per activar o desactivar els botons de la gestió de la reproducció. Els activarà o desactivarà d'acord amb els booleans existents al controlador.
- openRegWindow: Aquest mètode obrirà una finestra de tipus **FrmRegistreInici** perquè es puguin registrar nous usuaris o bé entrar-ne un de nou. Després cridarà els mètodes pubRepFill i privRepFill.
- openAddFileWindow: Aquest mètode obrirà una finestra de tipus **FrmAddMediaFile** perquè es pugui afegir un nou fitxer al repositori públic. Després cridarà els mètodes pubRepFill i privRepFill.
- emptyLists: Aquest mètode únicament farà que les llistes corresponents als dos repositoris siguin buides a l'inici de l'aplicació (abans que entri cap usuari). És cridat al main.
- privRepFill: Aquest mètode únicament farà visible la llista de fitxers existents al repositori privat de l'usuari. És cridat cada vegada que modifiquem alguna cosa del repositori, de manera que sempre la tinguem actualitzada en temps real.

- pubRepFill: Aquest mètode únicament farà visible la llista de fitxers existents al repositori públic. És cridat cada vegada que modifiquem alguna cosa del repositori, de manera que sempre la tinguem actualitzada en temps real.

L'aspecte gràfic que tindrà aquesta finestra serà el següent:

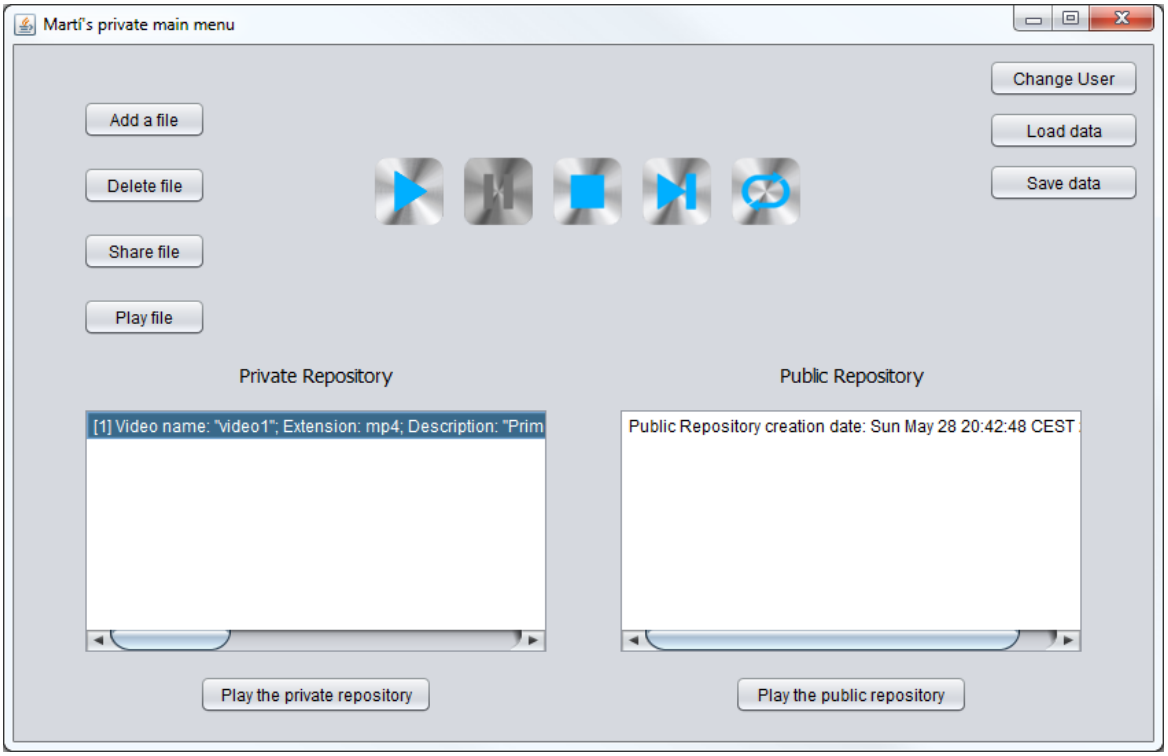

Figura 1.3: Aspecte gràfic que tindrà aquesta finestra.

Capítol 2

Altres qüestions

1. **Explica quin és el model de delegació d'esdeveniments que es fa servir en el botó de reproducció del repositori privat.**
   Vegem com s'afegeix un **ActionListener** al botó:

```
btnPlayPrivRep.setText("Play the private repository");
  btnPlayPrivRep.addActionListener(new java.awt.event.ActionListener() {
      public void actionPerformed(java.awt.event.ActionEvent evt) {
          btnPlayPrivRepActionPerformed(evt);
      }
  });
```

   Com podem veure, l'objecte on es produirà l'event, és a dir, la font d'event, serà el botó anomenat *btnPlayPrivRep*. El mètode per registrar el Listener serà addActionListener i l'objecte que gestionarà l'event serà *java.awt.event.ActionListener()*. Aleshores es cridarà el mètode btnPlayPrivRepActionPerformed, que rebrà l'event passat a actionPerformed i durà a terme les accions que he explicat anteriorment.

2. **Indiqueu quins tipus d'esdeveniments heu fet servir al vostre codi.**
   En el meu codi he fet servir diversos esdeveniments. Són els següents:

   - **ActionEvent:** He utilitzat aquest esdeveniment en moltes ocasions. Sempre que un botó duia a terme alguna acció era amb un esdeveniment d'acció, ja que amb l'acció de prémer el botó es cridava el mètode en qüestió.
   - **ItemEvent:** He utilitzat aquest esdeveniment per a veure un canvi en la selecció d'ítems d'un control de tipus "combo box".
   - **CaretEvent:** He utilitzat aquest esdeveniment per a veure un canvi en un determinat camp de text. L'he utilitzat a la classe **FrmAddMediaFile** per a modificar l'activació del botó d'acceptar.
   - **ListSelectionEvent:** He utilitzat aquest esdeveniment per a veure un canvi en la selecció de la llista de fitxers del repositori privat, a la classe **AppUB4**.

- **ChangeEvent:** He utilitzat aquest esdeveniment per a veure un canvi en l'estat del botó seleccionable de la reproducció cíclica, a la classe **AppUB4**.
- **MouseEvent:** He utilitzat aquest esdeveniment per a veure un moviment al ratolí. La meva intenció era tenir una funció que captés instantàniament quan acabava una reproducció, és a dir, que veiés com canviava l'estat del booleà de **Controller** que té aquesta informació. Però ho he implementat fent un mètode que actualitzés l'estat dels botons de reproducció cada vegada que es mogués el ratolí, és a dir, de manera quasi instantània.
- **WindowEvent:** He utilitzat aquest esdeveniment per a controlar les accions dutes a terme a l'obrir i tancar la finestra principal del menú de l'usuari. D'aquesta manera puc obrir una finestra per a registrar l'usuari just quan s'obre l'aplicació, i puc tancar correctament la finestra de reproducció abans de tancar la del menú.

3. Proves realitzades per comprovar el correcte funcionament de la pràctica, resultats obtinguts i accions derivades.

Separaré aquestes proves realitzades en les tres classes creades en aquesta entrega:

- **Classe FrmRegistreInici**: Les proves realitzades en aquesta finestra han sigut les següents:
  - Intentar afegir un mateix usuari dues vegades. Aquesta acció a l'entrega anterior em llançava una excepció. Ara, en aquesta, he hagut de crear un **JOptionPane** per notificar aquesta advertència.
  - Intentar entrar amb un usuari no registrat. Aquesta acció a l'entrega anterior em llançava una excepció. Ara, en aquesta, he hagut de crear un **JOptionPane** per notificar aquesta advertència.
  - Intentar tancar la finestra (evidentment sense haver entrat cap usuari, per la creueta de dalt a la dreta). Aquesta acció no tenia sentit, ja que no es pot accedir al menú principal sense haver entrat com a usuari. Per tant, he desactivat el fet que es pugui tancar la finestra. Només es podrà accedir al menú havent entrat un usuari registrat.
- **Classe FrmAddMediaFile**: Les proves realitzades en aquesta finestra han sigut les següents:
  - No ha estat ben bé una prova, pero he fet que els camps innecessaris per a certs fitxers estiguin desactivats quan es vol afegir aquest tipus de fitxers.
  - Intentar escriure una cadena de caràcters (lletres) en un camp on ha d'anar un *float* o un *int*. Al fer això, el mètode que afegia el fitxer no anava correctament. És per això que he cridat un mètode (parseFloat en cas dels *float*, parseInt en cas dels *int*) per a convertir aquestes *String* en el tipus de variable corresponent. No obstant això, si entrava un caràcter que no fos legal en algun d'aquests mètodes, s'aturava el programa. És per això que he decidit capturar l'excepció que llançava (*NumberFormatException*) i obrir una finestra de tipus **JOptionPane** mostrant el missatge d'error.
- **Classe AppUB4**: Les proves realitzades en aquesta finestra han sigut les següents:
  - Intentar fer servir un botó quan no es podia fer servir (per exemple, compartir un fitxer quan no n'hi havia cap de seleccionat, o pausar la reproducció quan no hi havia cap fitxer reproduint-se). En aquests casos, he desactivat tots els botons que no es poden

fer servir sense haver fet prèviament una acció. D'aquesta manera limito l'usuari i evito estar obrint constantment finestres d'error.

- Intentar eliminar diversos fitxers de cop. He hagut d'anar modificant l'índex d'eliminació per poder fer una eliminació simultània.
- Intentar reproduir dos fitxers seleccionats del repositori privat. He hagut de crear un **JOptionPane** per notificar a l'usuari que no es poden reproduir dos fitxers alhora.