# Programació 2 :
# Pràctica 3 - Central nuclear UB

*Alexander Borda Cuba*
*Oriol Morancho Negre*

**1. Expliqueu quines classes has pogut reutilitzar de la pràctica 3 part 1 per a fer la part 2.Quins canvis sobre les classes reutilitzades has necessitat fer i perquè.**

Les classes que hem reutilitzat en aquesta pràctica han estat totes les del directori model, així com l'Adaptador. Els pocs canvis en aquestes classes han estat relacionats amb l'accessibilitat a través de l'adaptador, com per exemple el dia: hem creat un mètode getDia() ja que no teníem accés a aquesta dada.

**2. Imagina que un company teu et passés la seva implementació de la classe Dades perquè la substituïssis en el teu codi. Quines modificacions serien necessàries en la vista i el adaptador? Com es relaciona la interfície InDades amb la pregunta anterior? Justifica la resposta.**

Tecnicament en aquestes practiques no caldria modificacions ni en vista ni en adaptador. El motiu es perque les practiques ens han proporcionat la interfície InDades, es a dir que seguim una pauta y si els dos ho hem implementat correctament no cal modificacions. Un altre cas es si el meu company no a fet el codi correctament, per exemple canvaint els noms que demana la interfície, llavors hauria de modificar l'adaptador que es el pont a vista, pero vista en cap dels casos.

**3. Quines conseqüències tindria que el mètode mostraIncidencies de la classe Adaptador retornés una llista d'objectes de la classe Incidencia en lloc d'un String?**

Si la funcio mostrarIncidencies retornés una llista es produiria un error perque a quan imprimeix la llista d'incidencies espera un String. La forma mes rapida d'arreglar aixo es afegint .toString(), que crida a la funcio de la llista imprimint els seus components en aquest format:

**[ objecte.toString(), objecte.toString(), …],**

si no hagessim definit el toString() de la classe Pagina Incidencies el resultat seria aquest:

**[prog2.model.PaginaIncidencies@4ca0981f, prog2.model.PaginaIncidencies@d26b9a9, prog2.model.PaginaIncidencies@b66cc10, prog2.model.PaginaIncidencies@2b56a976]**

**4. Indiqueu quins tipus d'esdeveniments heu fet servir al vostre codi.**

- **ChangeEvent — (via ChangeListener)**
  S'utilitza per detectar canvis en components com el JSlider. S'ha utilitzat en FrmGestioComponentsCentral.java i sldBarresControl (canvi de valor del percentatge de barres de control)

- **WindowEvent — (via WindowListener)**
  Captura esdeveniments de finestra com tancar la finestra. S'ha utilitzat a FrmVisualitzarInformacio.java i a addWindowListener(...) per gestionar el tancament de la finestra

**5. Explica quins canvis hauríeu de realitzar en l'aplicació si es volgués afegir la funcionalitat següent: En iniciar l'aplicació cal obrir una subfinestra en la qual l'usuari ha d'introduir el seu nom i aquest quedi registrat.**

La idea sería implementar una finestra semblant a la de guardar on es demanes el Nom de l'usuari y crear un atribut nom en Dades y un setter per a utilitzarlo mitjançant l'Adaptador.

**6. Observacions generals.**

Hem après a estructurar finestres principals i diàlegs. La reutilització de codi de la Part 1 va facilitar la feina, però ho hem hagut d'adaptar a la GUI. En general, la pràctica ens ha servit per consolidar els conceptes bàsics de programació orientada a esdeveniments i disseny d'interfícies.

**7. Diagrama de relacions entre les classes utilitzades.**