

Comentario general: los long serialVersionUID no sirven de nada, solo están para que no me aparezca con warnings (no relevantes) que me molestaba. Antes de entregar los quito. Te los recomienda el compilador. No lo vimos en teoría, ni te obligan a ponerlo.

## **Package Ventana**

### Ventana

Es un **JFrame** que crea los objetos **Panel\_1..4** y los configura con un [BoxLayout](#) con modelo de orden en vertical (**BoxLayout.Y\_AXIS**)

### Panel\_1

Es un **JPanel** con [GridBagLayout](#), se compone de un **JButton** y un **JProgressBar** y 3 **JLabel** (que son para rellenar espacios y poder centrar el botón en medio del panel)

- **GridBagConstraints: c** Es para poder configurar las posiciones de los elementos del panel con unos comandos (el .java tiene comentarios)
- + **actionPerformed**
  - Método que es llamado cuando se presiona el botón. El cual crea un objeto **ProcesarArchivos**.
- + **actualizarBarra**
  - Está para ser llamado por **Mensajero** y actualizar el valor del porcentaje de la **JProgressBar**.

### Panel\_2

Es un **JPanel** con [GridBagLayout](#) (podría ser un [BoxLayout](#) ahora que caigo) se compone por 3 **JPanel**, que a su vez contienen un **JTextField** y un **JLabel**.

- **Font: font** Crea un objeto [Font](#) (link para ver ejemplos), el cual es usado para configurar el tipo, tamaño y estilo de la letra.
- **configurarTextosyLabels**
  - Configura los **JTextField** para que el texto esté en el centro, y aplica el estilo de fuente en ellos y en los **JLabel**.
- **configurarSubPaneles**
  - Llama a subPanel1..3, los agrega y posiciona en el **JPanel** principal.
- **subPanel1..3**
  - Este método trabaja con un sub panel le crea un [BoxLayout](#) con modelo en orden en vertical se lo implanta, le configura que muestre los bordes mediante el [BorderFactory](#) y le agrega un **JTextField** y un **JLabel**.
- + **setText1..3**
  - Están para ser llamados por el **Mensajero** y cambiar el texto de los **JTextField**.

## Panel\_3

Es un **JPanel** con [BoxLayout](#), que se compone de otros 2 **JPanel** *panelSelector* y *panelTabla*.

- **configurarPanelSelector**
  - o Configura el *panelSelector* con [FlowLayout](#) y le agrega un **JLabel** y un **JComboBox**.
- **construyePanelTabla1..6**
  - o Crea un **DefaultTableModel** que es usado para configurar la **JTable**, poniendo los encabezados de las columnas y la cantidad de filas. Se le configura la fuente, de los encabezados (con `getTableHeader`) y de las celdas. Se le agrega el scroll a la **JTable**, mediante el **JScrollPane**. Para finalizar se le configura el Layout al panel correspondiente y se le agrega el **JScrollPane** (que ahora sería la tabla). Los pane
- **configurarPanelTabla**
  - o Crea un [CardLayout](#), que se le agrega al *panelTabla*, que servirá para desencadenar los eventos de cambio de panel se lo agrega al panel junto con todas las **JTable**.
- + **actionPerformed**
  - o Método que es llamado cuando se selecciona con el **JComboBox**. El cual, dependiendo del nombre vinculado al **JTable**, cambia de paneles.
- + **procesarTabla**
  - o Está para ser llamado por el **Mensajero**, y llama a cada **cargarTablaX** para rellene con los valores con los elementos de un **ArrayList**. El especial sería **cargarTablaTodos** porque pregunta si la cantidad de filas excede al que tenía configurado. El cual pregunta si la cantidad de elementos que quiero agregar supera el máximo ya impuesto y lo redimensiona. Yo a la *tablaTodos* le puse que tenga 1000 filas ya que es el máximo que nos da el enunciado y puede, o no, ser el máximo.

## Panel\_4

Es un **JPanel** con [BorderLayout](#). El panel tiene un título (no usé un **JLabel**), por lo cual se le tiene que poner un [BorderFactory](#) especial, al cual se le configura la posición y la fuente. Luego se le agrega un **JPanel** que contiene el dibujo.

- **PanelDibujo** Es una clase interna que se encarga de pintar el gráfico (el .java tiene comentarios)
- + **setAlturas** Está para ser llamado por el **Mensajero**, recibe un arreglo con las la cantidad de viewers. Itera y pone un límite por si exceder al límite del gráfico, luego hace un cálculo para ponerlo en escala.

## Package Rankigpeliculas

### Mensajero

Es un singleton. Se encarga de mandar elementos entre clases. Para ello necesita que cada Panel se mande para que el **Mensajero** los tenga de variable y así usar sus métodos públicos.

- + **mensajeBarra**
  - Llama al método **actualizarBarra** del **Panel\_1**.
- + **mensajeTotal**
  - Llama a los métodos **setText** del **Panel\_2**, enviándole los atributos usuario, movies y votos de la clase **Total**.
- + **mensajeTabla**
  - Llama al método **procesarTabla** del **Panel\_3**, enviándole un **ArrayList<FilaTabla>**.
- + **mensajeAlturas**
  - Crea un arreglo de int con los valores de los atributos rank0..5 de la clase **Total**, y se lo envía al **Panel\_4** usando el método **setAlturas**.

### ProcesarArchivos

Se encarga de procesar los archivos .csv y creando **ArrayList** con sus datos. Luego armar un árbol binario y armar otro **ArrayList** con orden descendente. Y para finalizar dispersarlo mediante el **Mensajero** a los **Paneles**.

- **procesarMovies**
  - Se encarga de procesar el archivo *movies.csv* (Está comentado en la clase)
- **procesarRatings**
  - Se encarga de procesar el archivo *ratings.csv* (Misma metodología anterior)
- **procesarLinea**
  - Recibe los datos de las líneas de ratings y los guarda en
- **crearArbol**
  - Recorre el **ArrayList** y crea el árbol binario ordenado por los viewers. Luego vacía el **ArrayList** para ser reutilizado por **procesarArbol**.
- **procesarArbol**
  - Recorre el árbol haciendo recorrido InOrden y agregando a un **ArrayList**, de modo que el arreglo esté ordenado de manera descendente.

## Package Objetos

### Total

Es una clase que sirve como contador.

### FilaTabla

Es un registro con los nombre, id, cantidad de viewers y el ranking de una película. Que representa las filas de las tablas del seleccionador.

- + **aumentarRanking**
  - Incrementa el ranking con uno recibido por parámetro.
- + **promedioRanking**
  - Hace el promedio entre la suma de todos los rankings previos con la cantidad de usuarios que votaron. La función **Math** es simplemente para redondear a un decimal.