86 -- 110 COMPONENTES\_Swing de java

VIDEO 86---**COMO CREAR CUADROS DE TEXTO; donde el usuario pueda pinchar y escribir**.

VIDEO 87-------------Añadimos más componentes a la lámina.

VIDEO 88---**Para saber cuando el usuario escribe, borra text, o saber de los cambios en el texto. (negrita, cursiva, tamaño, etc).**

VIDEO 90 **BARRA DE SCROOLL.Areas de texto, cuadroTex donde se puede poner varias lineas de texto. Utilizamos la cl JtextArea y sus mt.**

VIDEO 91 **s-------------** **TODO EL CODIGO DENTRO DE LA CLASE DEL MARCO VD-91**

,append(String texto), --agrega el texto de su parámetro cada vez que lo pulsamos.

VIDEO 92,------- **Casillas de verificación ----CHEKBOX**. Clase Font,

VIDEO 93 t-----------**,Botones de radio.**

VIDEO 94 -----CODIGO SIMPLIFICADO

VIDEO 95 **---ComboBox, para realizar desplegables en las interfacesGráficas**.

VIDEO 96 ---**Controles deslizantes o Jslider**,

VIDEO 97, --**Ejercicio con deslizantes o Jslider** -- 

VIDEO 98 --- **Control JSpinner**; cuadro de texto con dos botones ---

VIDEO 99

Cambiamos el incremento del botón que incrementa para que decremente y viceversa.

VIDEO 100----- Menus elemento fundamentas de una interface gráfica

fuente.addSeparator();// ------------para que aparecca una linea separando a 'opiones'

VIDEO 101 y 102------- EJERCICIO PRÁCTICO. Pone elementos a la escucha

VIDEO 104y 105**-------Código para que solamente cambie una de las propiedades del texto**.

VIDEO 106y 107**---StyledEditorKit** manejar el texto que hay dentro de un componente Swing de forma SENCILLA, -----**COLOCACION DE ICONOS EN LAS ETIQUETAS**

VIDEO 108 -- **-----------Menus con ‘CheckBox’ y ‘RadioButton’,**

VIDEO 109---**Menus Emergenes**, aparecen cuando pulsamos el botón derecho del ratón

VIDEO 110-------**Atajos de Teclado**, la cl ’**JmenuIten**’ dispone

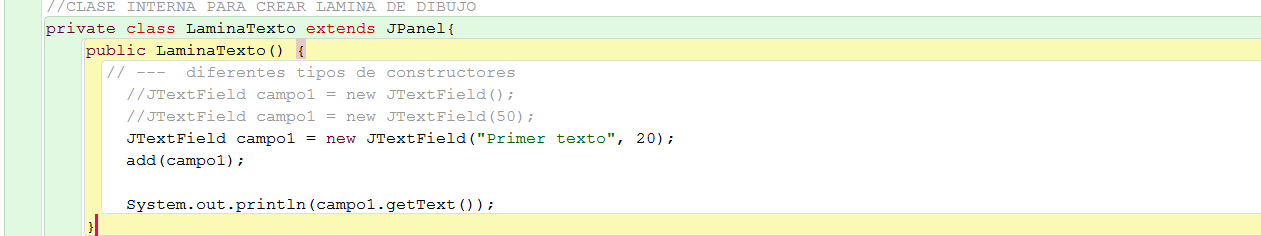


La cl JtextField construye el cuadro de texto en una única línea donde el usuario puede pulsar y escribir.

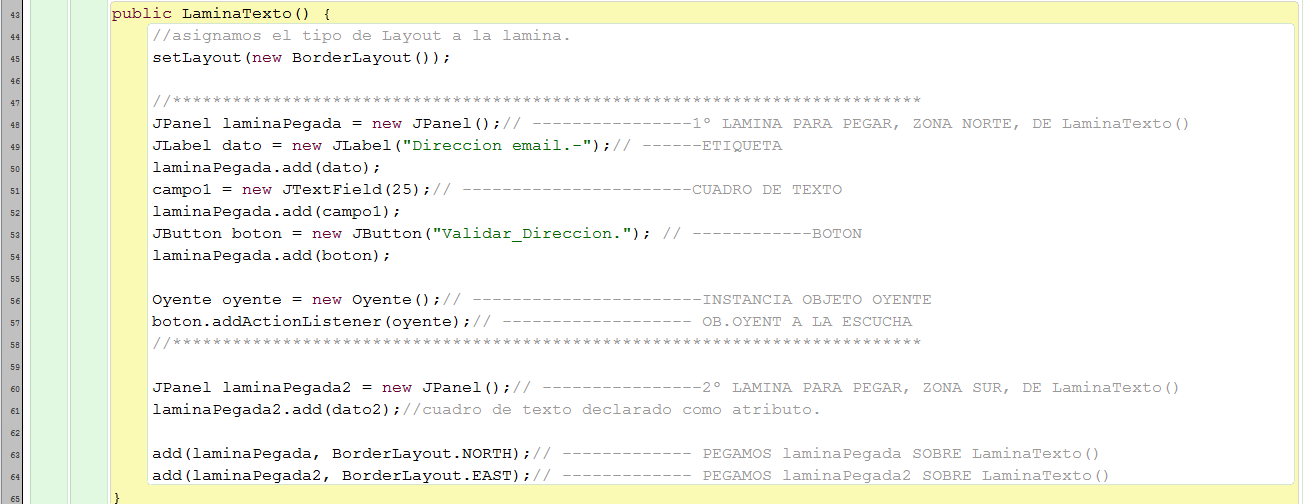
La cl JtextArea construye un area de texto que permite la construción de varias líneas.

Ambas cl heredan de la cl Abstracta JTextComponent, es la que contiene la mayoría de herramientas que usaremos como son los mt setText() y void getText();

Primer cuadro de texto dibujado en la lamina.

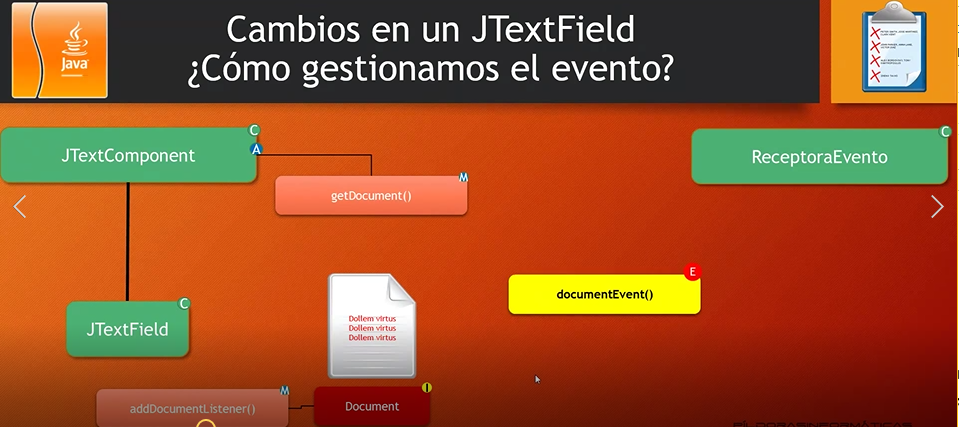


VIDEO 87 Añadimos más componentes a la lámina.

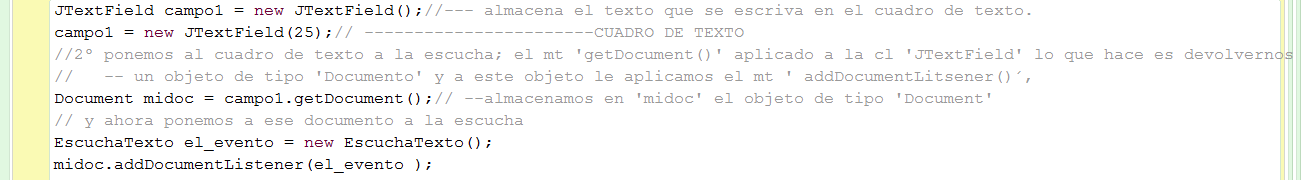


VIDEO 88

**Para saber cuando el usuario escribe, borra text, o saber de los cambios que puedan surgir en el texto. (negrita, cursiva, tamaño, etc)**



El mt getDocumet(), lo que hace es devolver un documento, o sea, ***un texto una representacion de lo qu hay dentro del JtexField, es ha este objetoDocumento al el que realiza el evento, no el CuadroDeTexto o JtextField,***



cuando invocamos getDocument(), se crea un objeto Document, el cual es una interface que tiene el mt addDocumentListener(), esto implica que si el cuadro de texto sufre alguna modificación, se dispara el evento, documentEvent(),

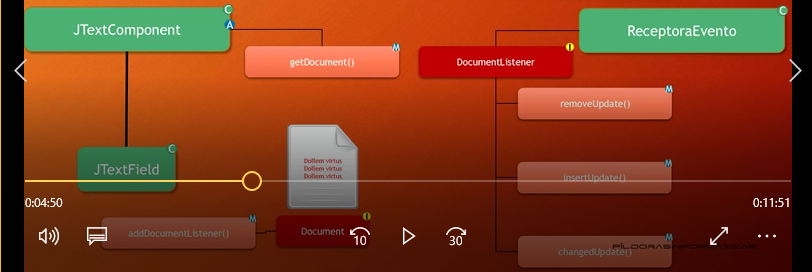
----\*\* Si tenemos una clase receptora de este tipo de eventos, ReceptoraEvento, podemos controlar estos eventos de tipo, documentEvent(),

---\*\* ReceptoraEvento, tiene que implementar la inteface, DocumentListener, la cual te obliga a sobreescribir sus tres mt, renoveUpdate(), insertUpdate(), changeUpdate(),

, renoveUpdate(), se dispara cuando dentro del cuadro de texto se elimina texto.

, insertUpdate(), se ejecuta cuando insertamos texto.

,changeUpdate(), se ejecuta cuando el texto que hay dentro de un cuadro modifica alguna de sus propiedades. (negrita, tamaño, etc)



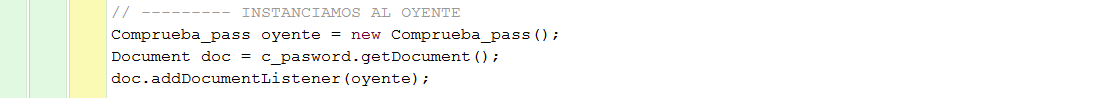
VIDEO 89

Pequeña aplicación gráfica de usuario que contiene una dirección\_email y el pasword que contenga entre 8 y 12 caracteres. El cuadroTx del pasword estará en rojo si no tiene entre 8 y 12 caracteres

CODIFICAMOS LA CLASE OYENTE.



Y donde hemos puesto los componentes a la lamina.



VIDEO 90

la Areas de texto, cuadroTex donde se puede poner varias lineas de texto. Utilizamos la cl JTextArea y sus mt.



,getText(), ---obtiene el texto.

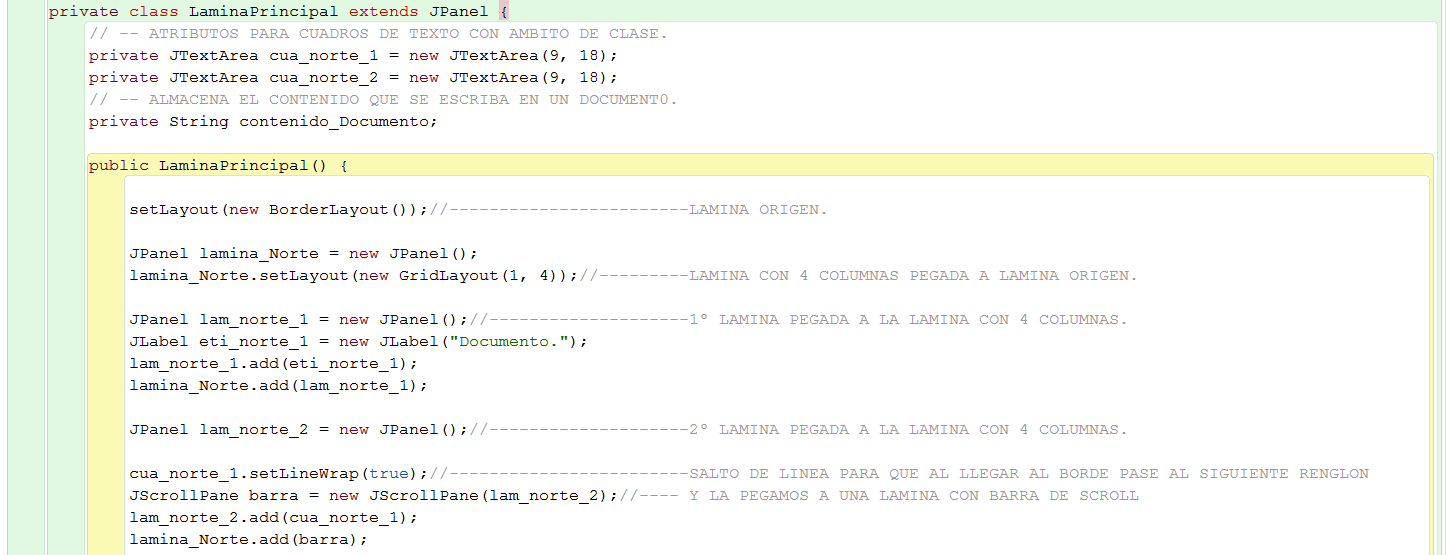
,setText(), --establece un texto en el area de texto.

,setLineWrap(boolean), --establece si un area de texto tiene saltos de línea o no los tiene, es decir, si el cuadro

De texto puede ensancharse o alargarse a medida que escribimos o no.

,getLineWrap(), --nos indica si el area de texto que estamos evaluando tiene o no saltos de línea.

, ,append(String texto), --agrega el texto de su parámetro cada vez que lo pulsamos.

**SOBRE LA BARRA DE SCROLL**.

1º Se crea una lámina con todos los componentes que precise, ‘lam\_norte\_1’.

2º Se precisa instanciar la cl ‘**JscrollPane**’ a la que se le pasa como parámetro, la lamina que llevará la barra, en , , este caso ‘lam\_norte\_1’.

3º el objetoLamina que se añade al marco o a la lámina original en este caso ‘lamina\_Norte’ es el objeto , , ‘**JscrollPane**  
**TAMBIEN DESTACA EL MT ‘’ .setLineWrap(true); ‘’’//------------------------***salto de linea para que al llegar al borde pase al siguiente renglon*

.

VIDEO 91

**.** ,getText(), ---obtiene el texto.

,setText(), --establece un texto en el area de texto.

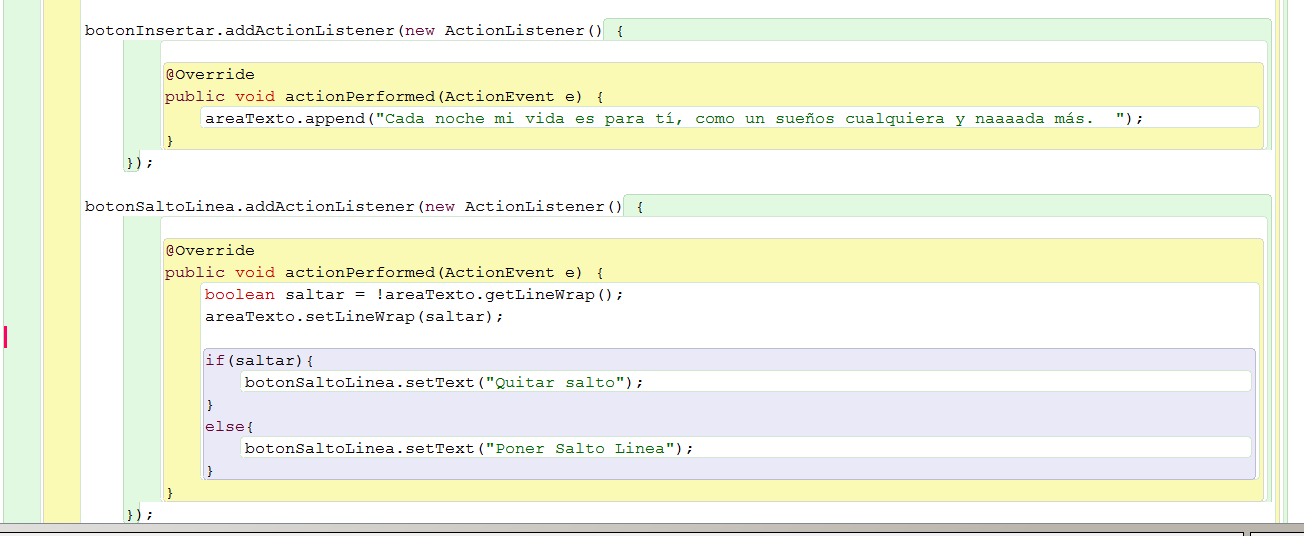
,setLineWrap(boolean), --establece si un area de texto tiene saltos de línea o no los tiene, es decir, si el cuadro

De texto puede ensancharse o alargarse a medida que escribimos o no.

,getLineWrap(), --nos indica si el area de texto que estamos evaluando tiene o no saltos de línea.

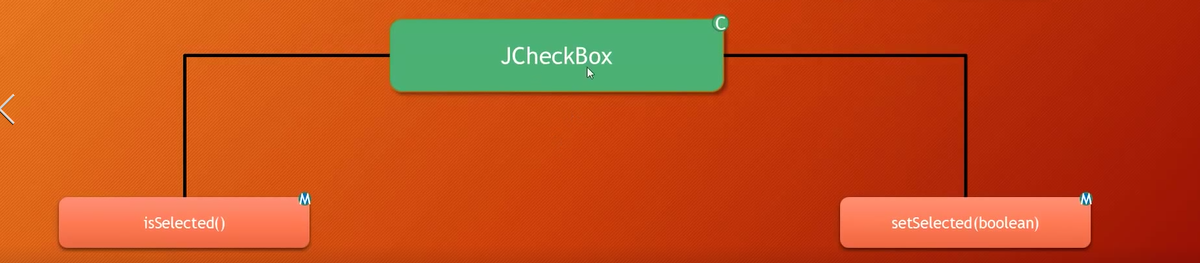
,append(String texto), --agrega el texto de su parámetro cada vez que lo pulsamos.

**Dentro de la cl Lamina**



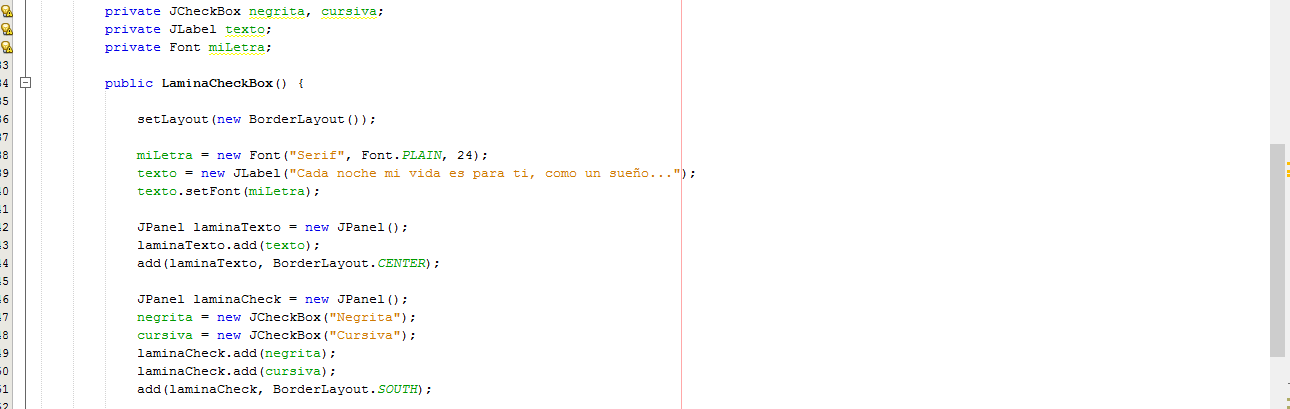
VIDEO 92

Casillas de verificación ----CHEKBOX.

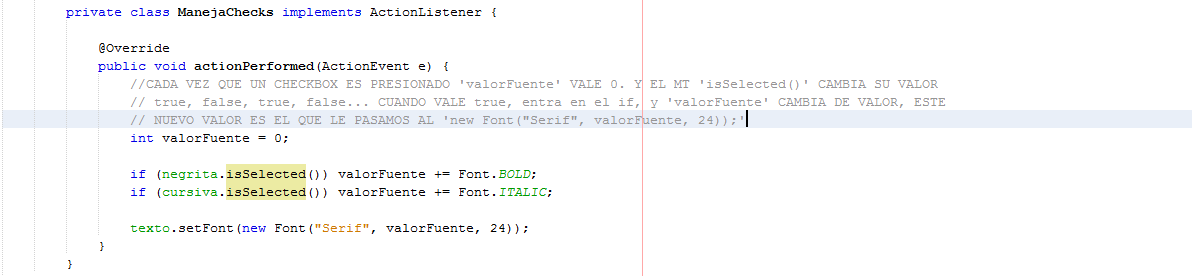


,isSelected(), determina si una casilla está seleccionada. Devuelve True o flse

,setSelected(boolean), permite seleccionar esa casilla, o no seleccionarla, establecer el estado de la casilla, si le pasas un true seleccionas esa casilla y con un false, quitas esa selección.



**Forma hábil de modificar un texto, utilizando los chekcsBox**



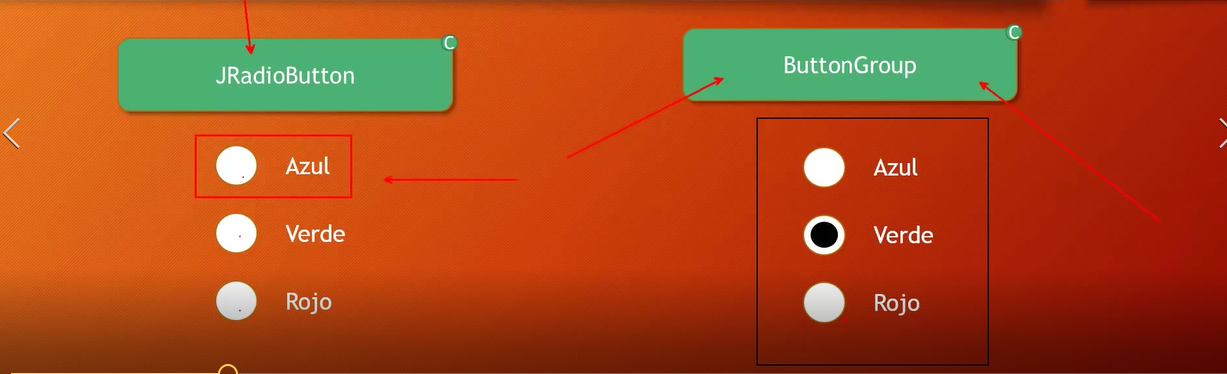
VIDEO 93

**Botones de radio.**

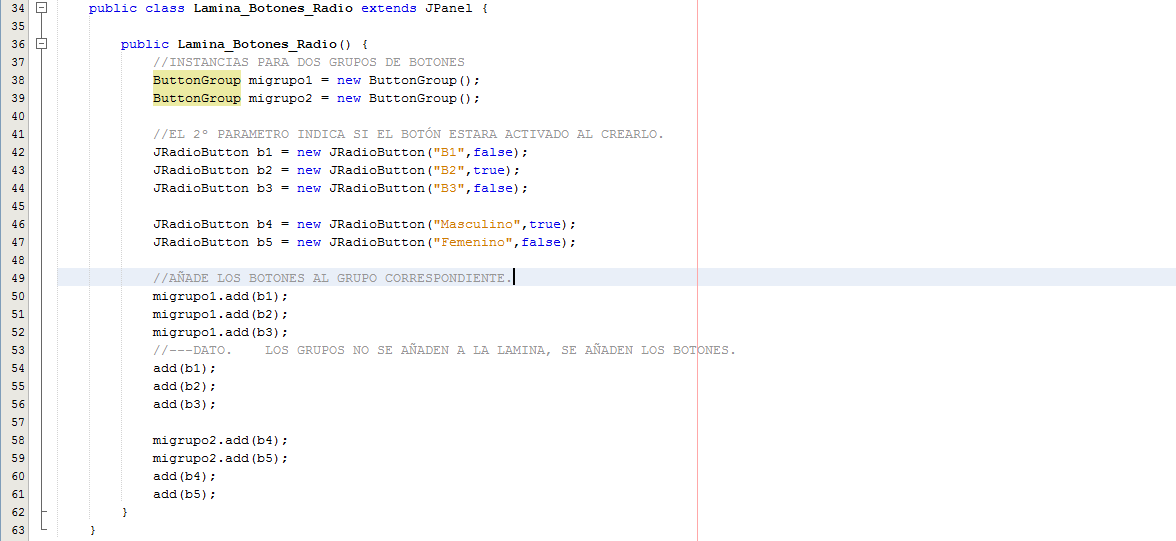
,Se utilizan para que el usuario pueda marcar una de las opciones que se le presenta, al contrario de lo que ocurria con los ChekcsBox, que el usuario puede seleccionar más de una opción.

,JRadioButton, con esta clase se crea cada uno de los botones.

,ButtonGroup, agrupa a varios botones dentro de un mismo grupo, cuando están seleccionados es cuando se consigue que solo uno de ellos pueda estar seleccionado.

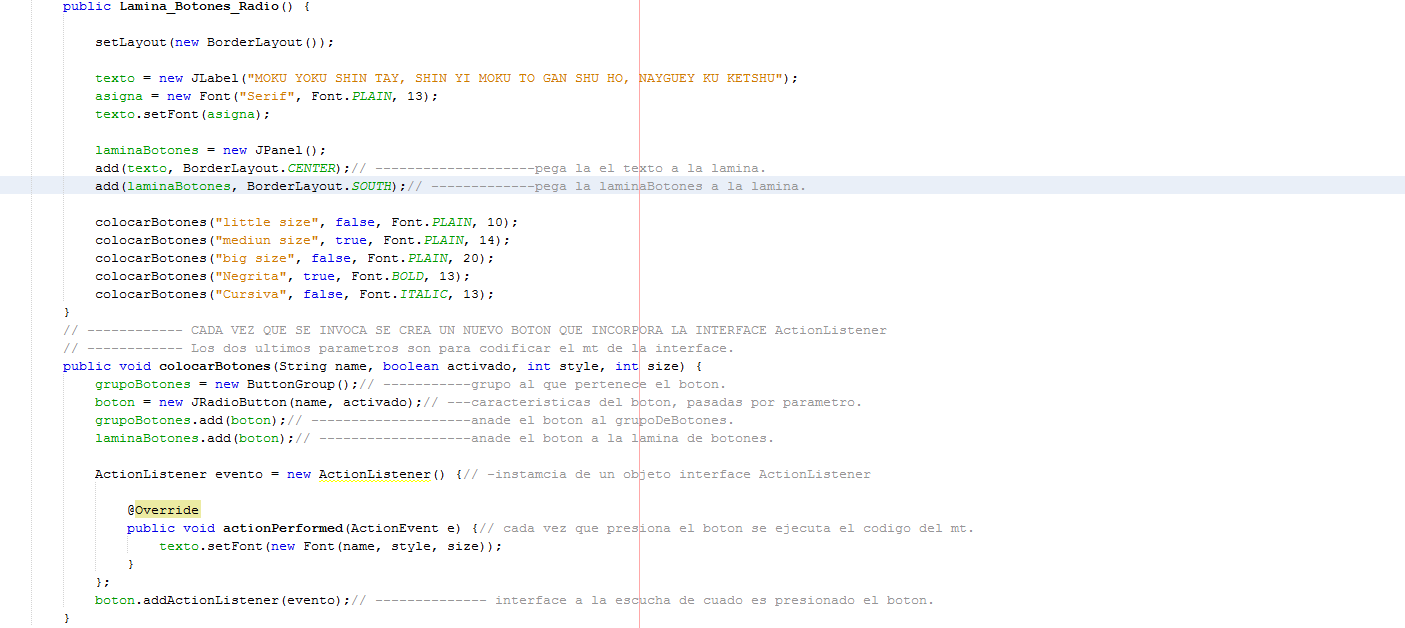


//---DATO LOS **BOTONES-GRUPO** NO SE AÑADEN A LA LAMINA, SE AÑADEN LOS BOTONES.



VIDEO 94

CODIGO SIMPLIFICADO



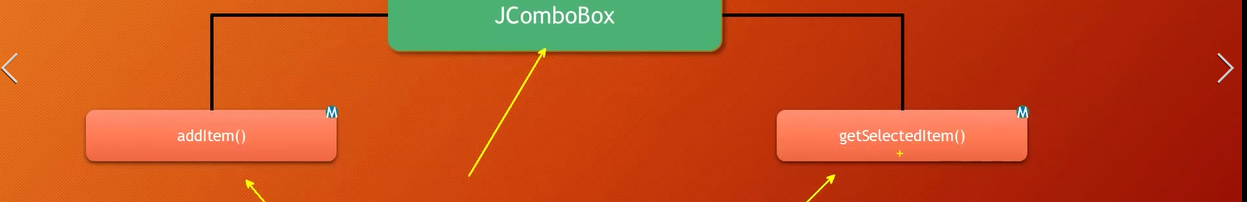
VIDEO 95

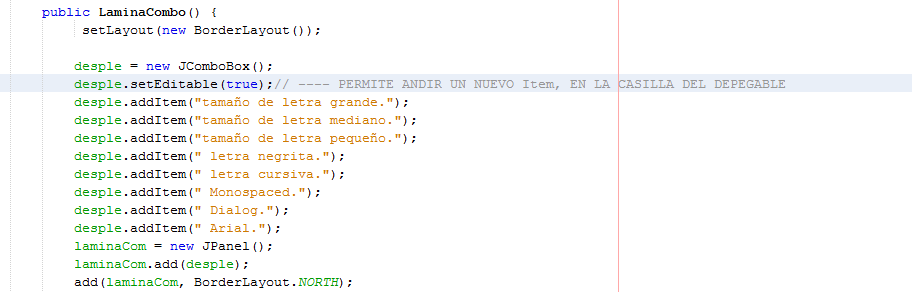
**ComboBox, para realizar desplegables en las interfacesGráficas**.

,addItem(), permite añadir un enunciado en el despegable.

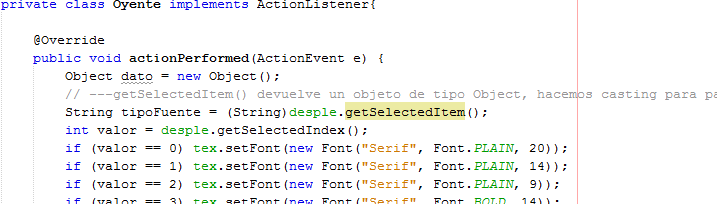
,getSelectedItem(), permite saber el desplegable seleccionado por el usuario.

, setEditable(true); // ---- PERMITE ANDIR UN NUEVO Item, EN LA CASILLA DEL DEPEGABLE





**Interface**



VIDEO 96 ---

**Controles deslizantes o Jslider**,

Es el tipico control que puedes arrastrar arriba/ abajo o a los lados, (Subir volumen, aumentar tamaño)



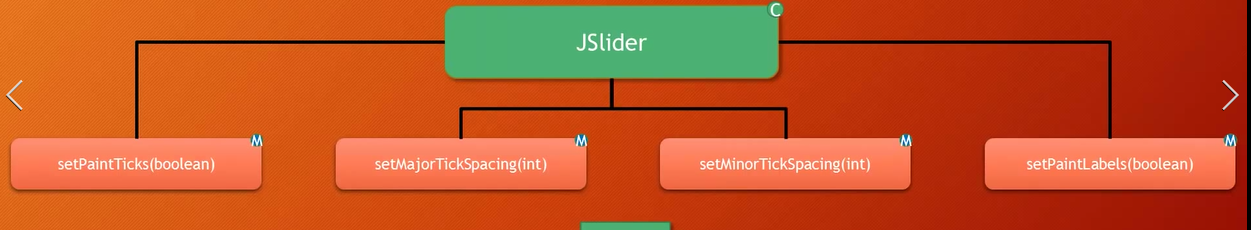
,setPaintTicks(boolean); --- para que salgan las líneas,

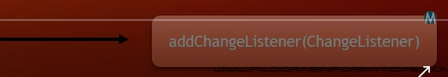
,setMajorTickSpacing(int), -- determinas de cuanto en cuanto van las marcas mayores.

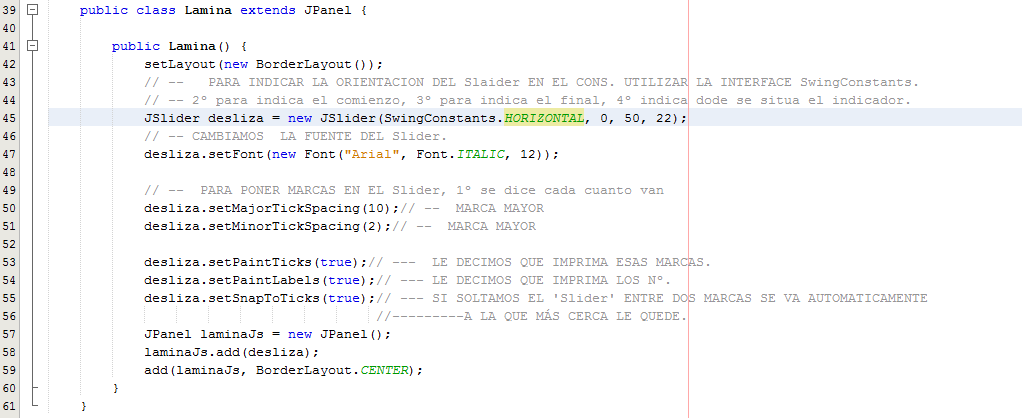
,setMinorTickSpacing(int), -- determinas de cuanto en cuanto van las marcas menores.

,setPaintLabels(boolean); ---para que salgan los valores numéricos.

,



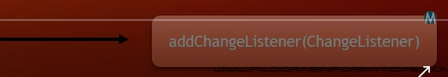
addChangeListener(ChangeListener), -- pone el componente a la escucha de un cambio en su estado , el parámetro es una interface ‘ChangeListener’  . Cada vez que movemos el Slaider de un lado a otro lo que hacemos es propagar un objeto de tipo ‘ChangeEvent e’ , que recoge el mt ‘stateChanged(…)’ de la interface ChangeListener’



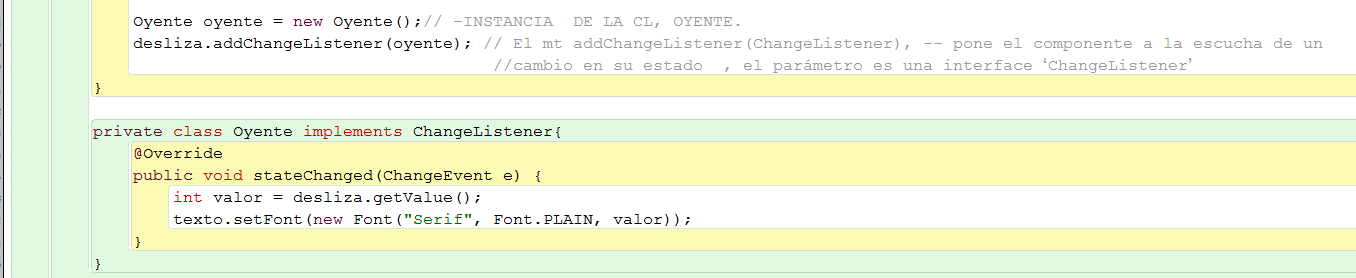
VIDEO 97, --

**Ejercicio con deslizantes o Jslider**

El mt addChangeListener(ChangeListener), -- pone el componente a la escucha de un cambio en su estado , el parámetro es una interface ‘ChangeListener’



 Cada vez que movemos el Slaider de un lado a otro lo que hacemos es propagar un objeto de tipo ‘ChangeEvent e’ , que recoge el mt ‘stateChanged(…)’ de la interface ChangeListener’



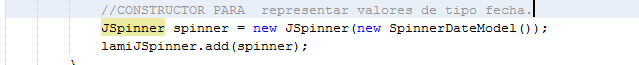
VIDEO 98

--- **Control JSpinner**; es un cuadro de texto con dos botones a la derecha, ***normalmente tiene datos de tipo número, pero pueden tener fechas, listas de cosas etc***, para con estos botones incrementar o disminuir esos valores numéricos. Utilizamos la cl Jspiner.

El **constructo**r con el parámetro de tipo, **SpinnerModel**, permite representar otros valores que no sean numéricos.

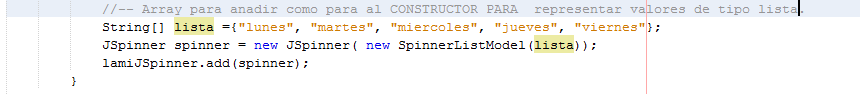


Si quisieramos representar valores de tipo fecha, tendriamos que pasarle como parámetro un objeto de la cl ‘SpinnerDateModel’ si de tipo lista sería la cl ‘SpinnerListModel’

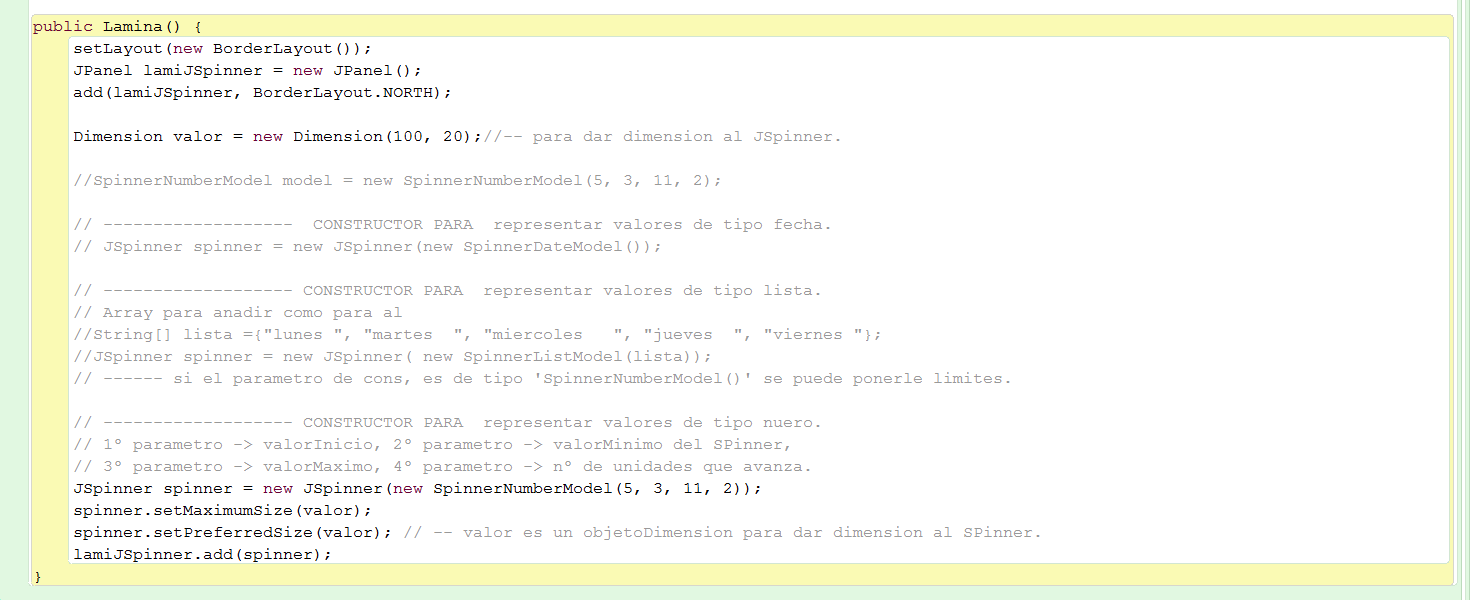




**Y ,**





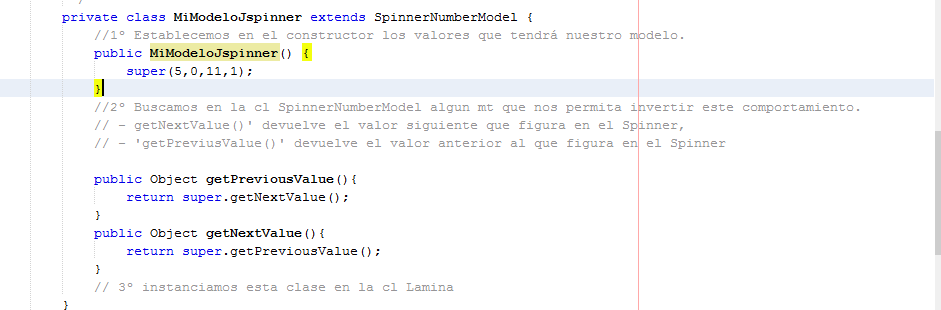


VIDEO 99

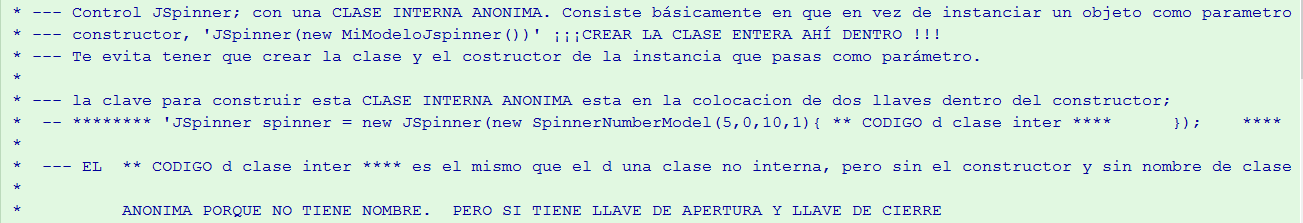
Cambiamos el incremento del botón que incrementa para que decremente y viceversa.

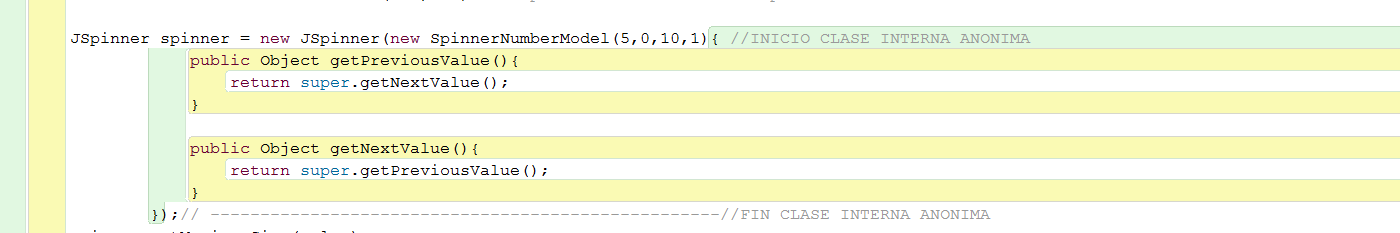
,para ello tenemos que crear nuestro propio modelo, SpinnerDateModel’ o ‘SpinnerListModel’ o SpinnerNumberModel’ no sirven.

Tenemos que crear una clase para este modelo



CON UNA CLASE INTERNA ANONIMA





VIDEO 100

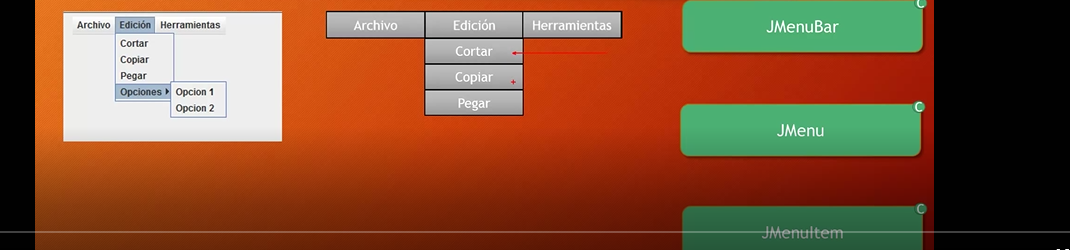
U------- Menus elemento fundamentas de una interface gráfica

1º crear barra, sera el soporte, utilizamos la cl ,JMenuBar,

2º los elementos que forman el menu, cl ,JMenu, cada vez que instancias esta cl, creas un elemento de menu, que estarán dentro de la barra de menú.



3º de algunos de estos elementos de menu colgaran algunas opciones, para ello está la cl ,JMenuItem, cada vez que instancias esta cl, creas un elemento.





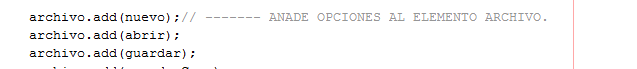
2º



3º



4º

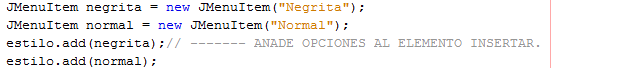


5º

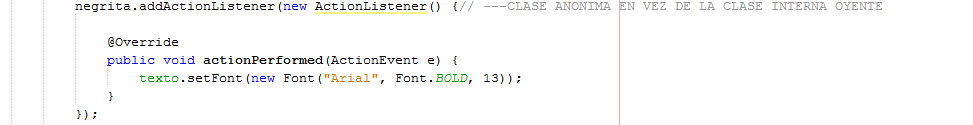


VIDEO 101 y VIDEO 102

EJERCICIO PRÁCTICO. Pone elementos a la escucha



con una clase interna anónima.

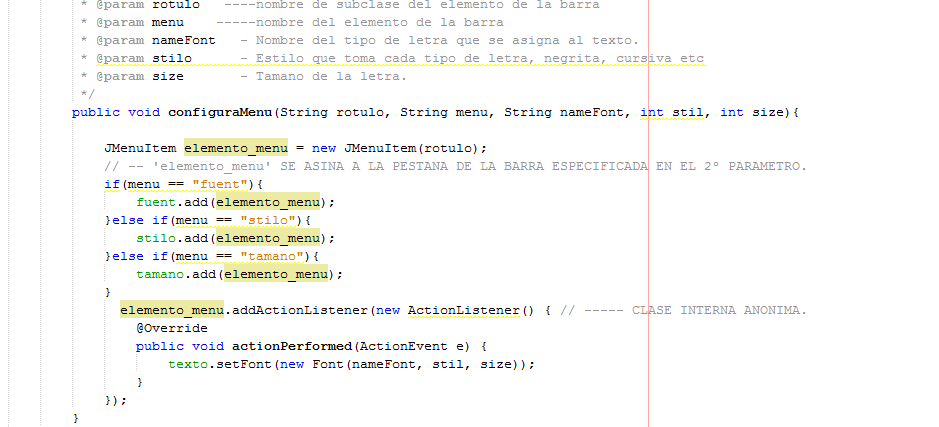


EJERCICIO PRÁCTICO. Pone elementos a la escucha

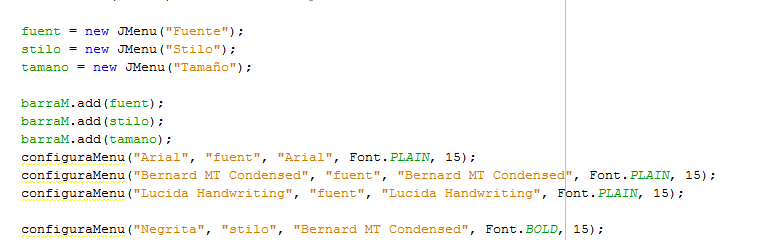
VIDEO 103

**REDUCIDO CODIGO.**

SE CREAN TODOS LAS SUBCLASES DE LOS ELEMENTO DE LA BARRA EN UN MÉTODO.



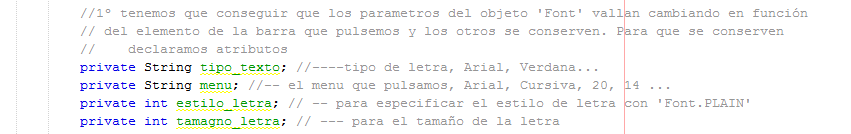
SE ASIGNAN A CADA PESTAÑA.

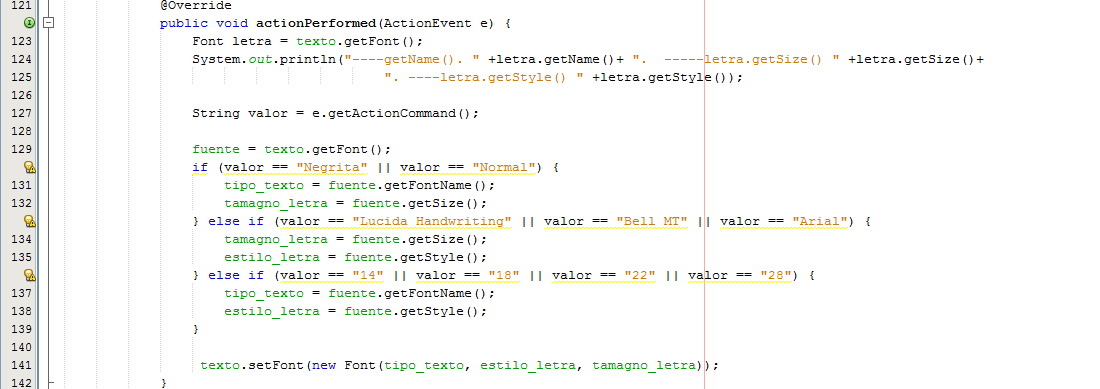


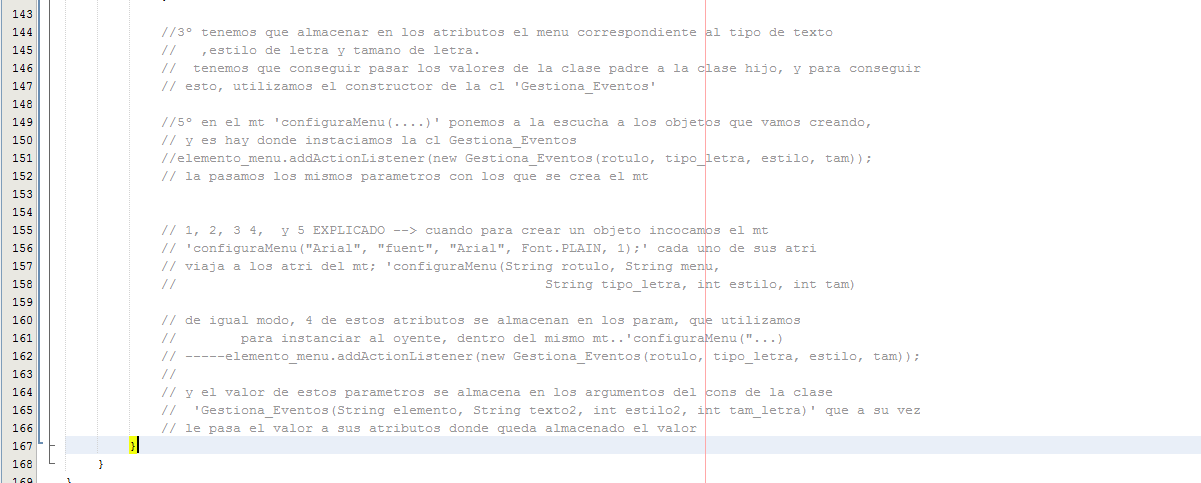
VIDEO 104y 105

**Código para que solamente cambie una de las propiedades del texto**.

Que solo cambie el estilo de letra, el tamaño o la fuente.







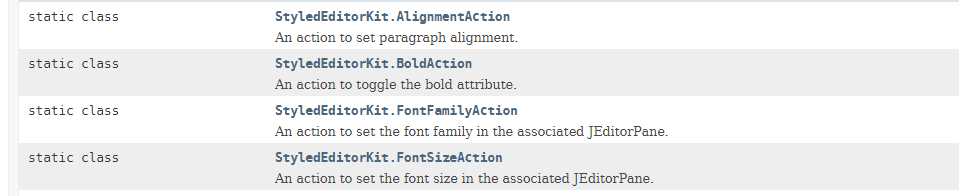
VIDEO 106

**StyledEditorKit** Clase paraa manejar el texto que hay dentro de un componente Swing de forma muy sencilla.

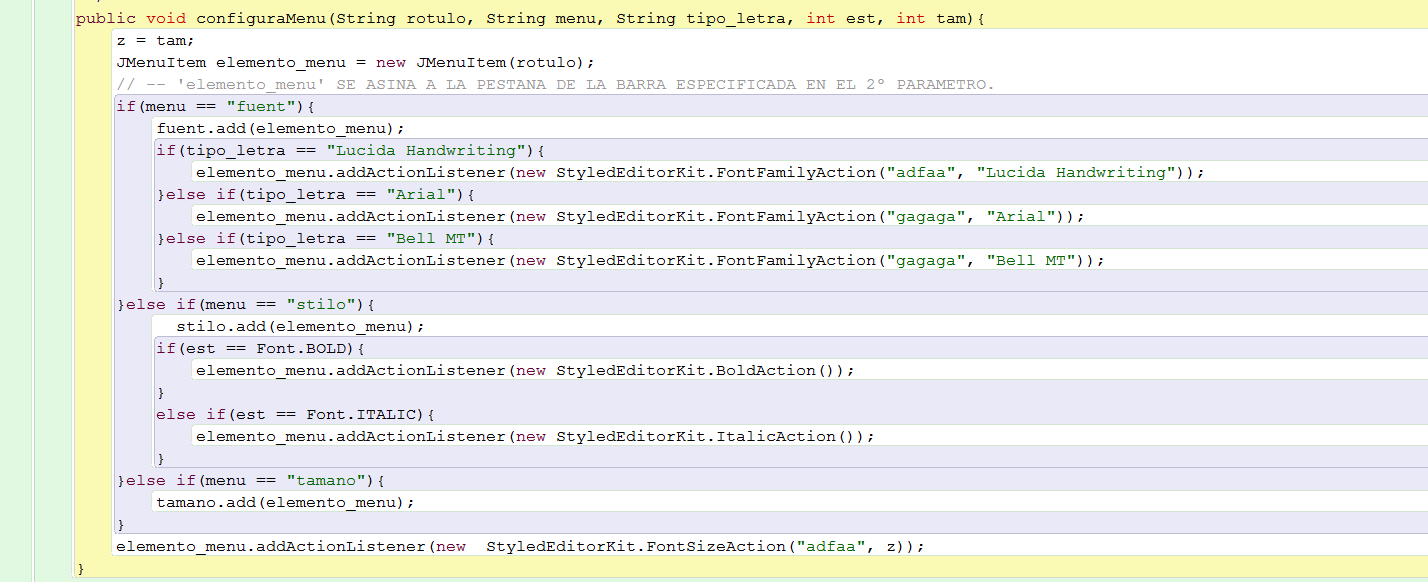
1º permite cambiar las caracteristicas del texto seleccionado.

2º quitar la negrita o la cursiva si pulsamos una segunda vez.

En la api vemos que dispone de varias cl, internas.



De facil utilización.



VIDEO 107

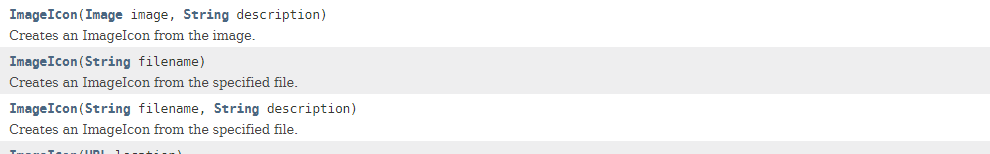
**StPoner iconos en las etiquetas** -----

Utilizamos el constructor adecuado. ‘[**JMenuItem**](file:///C:\Users\Usuario\Desktop\CURSO_JAVA\00%20Todas_Las_Clases_De_Java\api\javax\swing\JMenuItem.html#JMenuItem-java.lang.String-javax.swing.Icon-)([**String**](file:///C:\Users\Usuario\Desktop\CURSO_JAVA\00%20Todas_Las_Clases_De_Java\api\java\lang\String.html) text, [**Icon**](file:///C:\Users\Usuario\Desktop\CURSO_JAVA\00%20Todas_Las_Clases_De_Java\api\javax\swing\Icon.html) icon)’

El parámetor ‘Icon icon’ vemos en la api de java que;

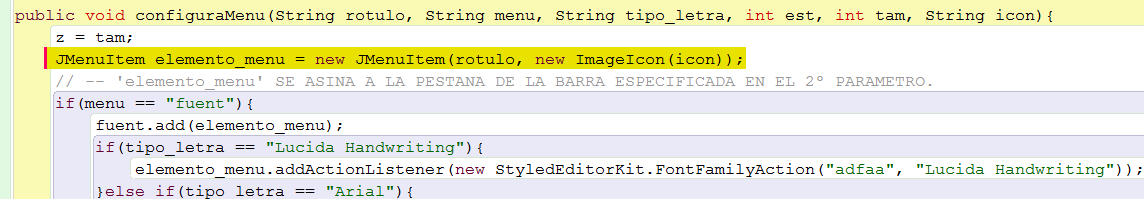


Implementa la cl, ‘ImageIcon’, la cual tiene sobrecarga de constructores,



Utilizando uno de ellos, instanciamos directamente la clase ‘Image’ dentro del constructor del ‘JmenuItem’,

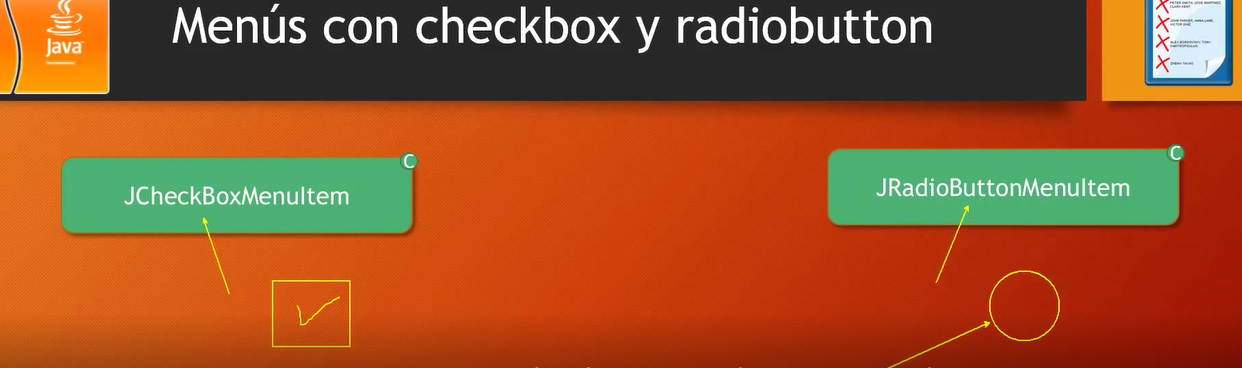
JMenuItem elemento\_menu = new JMenuItem(rotulo, new ImageIcon(icon));



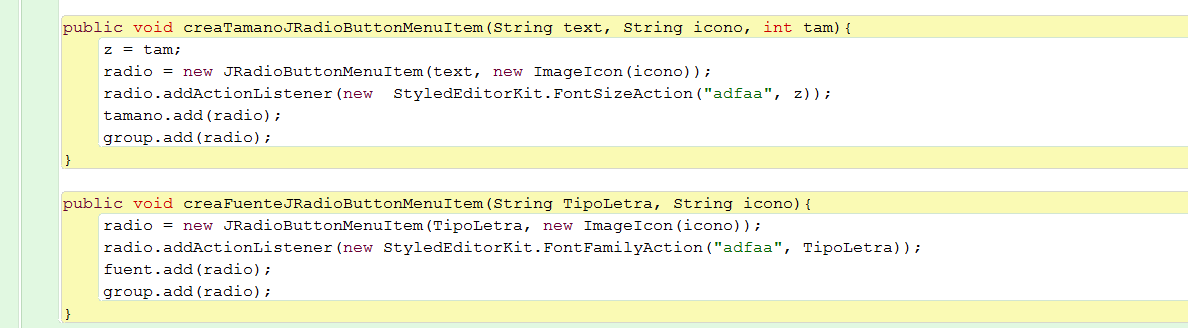
VIDEO 108

---Menus con ‘CheckBox’ y ‘RadioButton’, los primeros pueden seleccionarse varios, los RadioButton en cambio solo se puden seleccionar uno de ellos, (*solo se puede ser de género masculino, o femenino*).

Para aplicarlos a menus tenemoss que utilizar las siguientes cl.



Crea nuevos mt para poner ‘CheckBox’ y ‘RadioButton’ a las etiquetas. El mt configurar menú, se elimina.



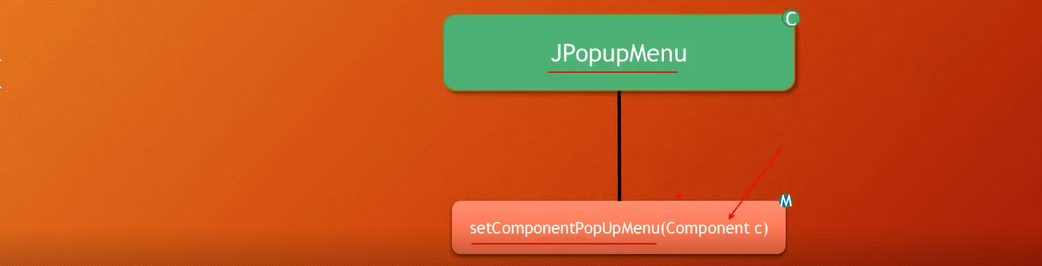
Y les invocamos en el constructor de la Lamina.



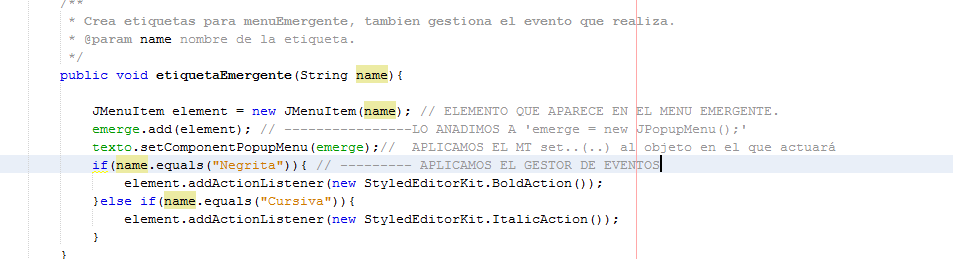
Y 

VIDEO 109

---**Menus Emergenes**, aparecen cuando pulsamos el botón derecho del ratón. Para coger de lo que aparece en ese menú lo que se quiera.



Tenemos que utilizar la cl ‘JPopupMenu’ y un mt que herda de la cl ‘Component’, ‘setComponentPopUpMenu(Component c)’, nos pide un parámetro que hace referencia al elemento sobre el que se aplica el mt, no es os mismo un ‘JTextArea’ que un ‘JButton’, o sea, donde haremos click con el botón derecho para que salga el menú emergente.



LUEGO INVOCAMOS EL MT EN EL CONS, DE LA CL,



VIDEO 110

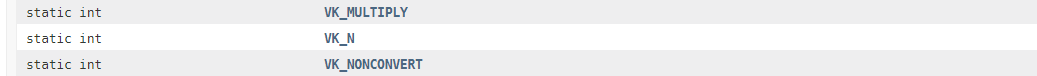
-------Atajos de Teclado, la cl ’**JmenuIten**’ dispone del mt, ‘**setAccelerator(KeyStroke keyStrokd)**’ para realizar acciones de teclado.



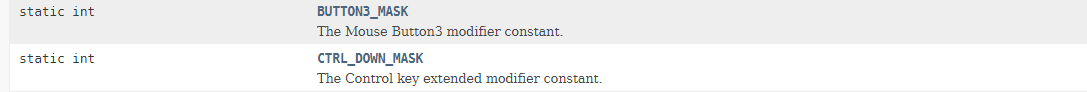
## Pide como parámetro un objeto de tipo ‘KeyStroke’ y en esta cl tenemos el mt;



El parámetro **KeyCode** - especifica el código numérico para una clave del teclado, cada tecla tiene un nº asignad



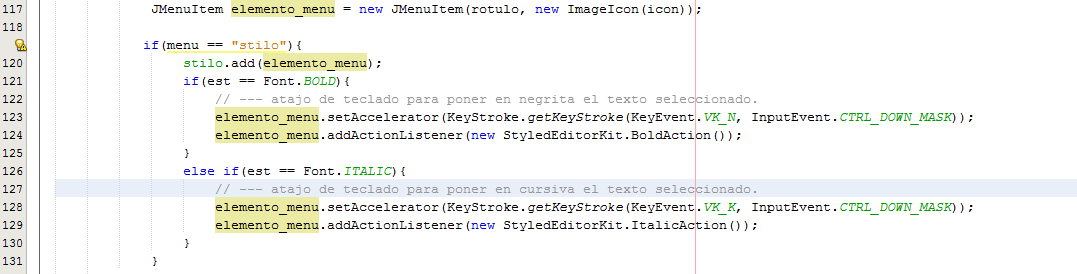
El parám **Modificadores** - una combinación de cualquier modificador (Ctrl, Alt, Ins …) tiene un nº asignado.



Para ver la informacion de estas variables tenemos que ver la información que aporta el mt, getKeyStroke al final.en

. 

**Aplicado al código;**



.

.