

TIPOS DE DATOS PRIMITIVOS EN JAVA				
Tipo	Valores	Por defecto	Tamaño	Rango
byte	enteros con signo	0	8 bits	-128 a 127
short	enteros con signo	0	16 bits	-32768 a 32767
int	enteros con signo	0	32 bits	-2147483648 a 2147483647
long	enteros con signo	0	64 bits	-9223372036854775808 a 9223372036854775807
float	IEEE 754 punto flotante	0.0	32 bits	$\pm 1.4E-45$ a $\pm 3.4028235E+38$ y $\pm\infty, \pm 0, \text{NaN (Not A Number)}$
double	IEEE 754 punto flotante	0.0	64 bits	$\pm 4.9E-324$ a $\pm 1.7976931348623157E+308$ y $\pm\infty, \pm 0, \text{NaN (Not A Number)}$
char	Carácter Unicode	\u0000	16 bits	\u0000 a \uFFFF
boolean	true, false	false	Entero de 32 bits (sólo usa un bit)	

MENÚ EN NETBEANS
<ol style="list-style-type: none"> 1. File: Creación y apertura de elementos: proyectos, clases, interfaces, etc. 2. Edit: comandos típicos de edición de texto 3. View: muestra u oculta elementos de la ventana del editor (p. ej. n^{os} de línea del editor) 4. Navigate: desplazamientos, búsquedas y reemplazos en el código 5. Source: más opciones de edición de código (duplicar o desplazar líneas, p. ej.), solución de errores sintácticos, etc. 6. Refactor: modificación de código (p. ej. cambio de nombre de clases, modificación en el n^o de argumentos de un método, etc.) 7. Build: compilación de código 8. Run: ejecución y depuración 9. Profile: análisis del código 10. Versioning: control de versiones 11. Tools: <ol style="list-style-type: none"> a. configuración (versión de java a usar, librerías a incluir, etc. b. herramientas externas varias: testeo, xml, etc. 12. Window: organización de las ventanas del editor 13. Help

CREACIÓN DE UNA APLICACIÓN SWING SIMPLE DE MODO AUTOMÁTICO CON NETBEANS
<ol style="list-style-type: none"> 1. File → Nuevo Proyecto → Categoría: java / Proyecto: java application → Next 2. Indicar el nombre y ubicación de la aplicación (proyecto). Indicar que no debe crear clase main 3. File → Nuevo Fichero → Categoría: Formularios GUI Swing / Tipo de fichero: JFrame 4. Indicar el nombre de la ventana. Se podría indicar también el paquete de la ventana. 5. En la ventana de diseño se añaden botones, etiquetas, etc. de la paleta (a la derecha) 6. Modificar las propiedades de los controles (texto, colores, etc.) en la ventana de propiedades (dcha.) 7. Cuando se quiera escribir el código para un evento concreto, se debe seleccionar el control, y en la ventana de propiedades activar la pestaña de <i>Eventos</i>. En ella, hacer clic en la celda a la derecha del evento de interés. Se abrirá una función en la ventana de código, donde se escribirá la respuesta al evento.

LISTA DE ESCUCHADORES Y ADAPTADORES DE EVENTOS

Más información en: http://blogs.sun.com/CoreJavaTechTips/entry/listeners_vs_adapters

Listener Interface	Adapter Class	Listener Methods
ActionListener	<i>none</i>	actionPerformed(ActionEvent)
AncestorListener	<i>none</i>	ancestorAdded(AncestorEvent) ancestorMoved(AncestorEvent) ancestorRemoved(AncestorEvent)
CaretListener	<i>none</i>	caretUpdate(CaretEvent)
CellEditorListener	<i>none</i>	editingStopped(ChangeEvent) editingCanceled(ChangeEvent)
ChangeListener	<i>none</i>	stateChanged(ChangeEvent)
ComponentListener	ComponentAdapter	componentHidden(ComponentEvent) componentMoved(ComponentEvent) componentResized(ComponentEvent) componentShown(ComponentEvent)
ContainerListener	ContainerAdapter	componentAdded(ContainerEvent) componentRemoved(ContainerEvent)
DocumentListener	<i>none</i>	changedUpdate(DocumentEvent) insertUpdate(DocumentEvent) removeUpdate(DocumentEvent)
ExceptionListener	<i>none</i>	exceptionThrown(Exception)
FocusListener	FocusAdapter	focusGained(FocusEvent) focusLost(FocusEvent)
HierarchyBoundsListener	HierarchyBoundsAdapter	ancestorMoved(HierarchyEvent) ancestorResized(HierarchyEvent)
HierarchyListener	<i>none</i>	hierarchyChanged(HierarchyEvent)
HyperlinkListener	<i>none</i>	hyperlinkUpdate(HyperlinkEvent)
InputMethodListener	<i>none</i>	caretPositionChanged(InputMethodEvent) inputMethodTextChanged(InputMethodEvent)
InternalFrameListener	InternalFrameAdapter	internalFrameActivated(InternalFrameEvent) internalFrameClosed(InternalFrameEvent) internalFrameClosing(InternalFrameEvent) internalFrameDeactivated(InternalFrameEvent) internalFrameDeiconified(InternalFrameEvent) internalFrameIconified(InternalFrameEvent) internalFrameOpened(InternalFrameEvent)
ItemListener	<i>none</i>	itemStateChanged(ItemEvent)
KeyListener	KeyAdapter	keyPressed(KeyEvent) keyReleased(KeyEvent) keyTyped(KeyEvent)
ListDataListener	<i>none</i>	contentsChanged(ListDataEvent) intervalAdded(ListDataEvent) intervalRemoved(ListDataEvent)
ListSelectionListener	<i>none</i>	valueChanged(ListSelectionEvent)
MenuDragMouseListener	<i>none</i>	menuDragMouseDragged(MenuDragMouseEvent) menuDragMouseEntered(MenuDragMouseEvent) menuDragMouseExited(MenuDragMouseEvent) menuDragMouseReleased(MenuDragMouseEvent)
MenuKeyListener	<i>none</i>	menuKeyPressed(MenuKeyEvent) menuKeyReleased(MenuKeyEvent) menuKeyTyped(MenuKeyEvent)
MouseListener	<i>none</i>	menuCanceled(MenuEvent) menuDeselected(MenuEvent) menuSelected(MenuEvent)
MouseInputListener (extends MouseListener and MouseMotionListener)	MouseInputAdapter MouseAdapter	mouseClicked(MouseEvent) mouseEntered(MouseEvent) mouseExited(MouseEvent) mousePressed(MouseEvent) mouseReleased(MouseEvent) mouseDragged(MouseEvent)

		mouseMoved(MouseEvent) MouseListener(MouseEvent)
MouseListener	MouseListener, MouseListenerAdapter	mouseClicked(MouseEvent) mouseEntered(MouseEvent) mouseExited(MouseEvent) mousePressed(MouseEvent) mouseReleased(MouseEvent)
MouseListenerAdapter	MouseListenerAdapter, MouseListenerAdapter	mouseDragged(MouseEvent) mouseMoved(MouseEvent)
MouseWheelListener	MouseListenerAdapter	mouseWheelMoved(MouseWheelEvent) MouseListenerAdapter
PopupMenuListener	<i>none</i>	popupMenuCanceled(PopupMenuEvent) popupMenuWillBecomeInvisible(PopupMenuEvent) popupMenuWillBecomeVisible(PopupMenuEvent)
PropertyChangeListener	<i>none</i>	propertyChange(PropertyChangeEvent)
TableModelListener	<i>none</i>	columnAdded(TableModelEvent) columnMoved(TableModelEvent) columnRemoved(TableModelEvent) columnMarginChanged(ChangeEvent) columnSelectionChanged(ListSelectionEvent)
TableModelListener	<i>none</i>	tableChanged(TableModelEvent)
TreeExpansionListener	<i>none</i>	treeCollapsed(TreeExpansionEvent) treeExpanded(TreeExpansionEvent)
TreeModelListener	<i>none</i>	treeNodesChanged(TreeModelEvent) treeNodesInserted(TreeModelEvent) treeNodesRemoved(TreeModelEvent) treeStructureChanged(TreeModelEvent)
TreeSelectionListener	<i>none</i>	valueChanged(TreeSelectionEvent)
TreeWillExpandListener	<i>none</i>	treeWillCollapse(TreeExpansionEvent) treeWillExpand(TreeExpansionEvent)
UndoableEditListener	<i>none</i>	undoableEditHappened(UndoableEditEvent)
VetoableChangeListener	<i>none</i>	vetoableChange(PropertyChangeEvent)
WindowFocusListener	WindowAdapter	windowGainedFocus(WindowEvent) windowLostFocus(WindowEvent)
WindowListener	WindowAdapter	windowActivated(WindowEvent) windowClosed(WindowEvent) windowClosing(WindowEvent) windowDeactivated(WindowEvent) windowDeiconified(WindowEvent) windowIconified(WindowEvent) windowOpened(WindowEvent)
WindowStateListener	WindowAdapter	windowStateChanged(WindowEvent)