

# **ENTORNO DE PROGRAMACION OPEN SOURCE**

## **NETBEANS V. 4.1**

### **INTRODUCCIÓN AL TUTORIAL**

Esta guía esta diseñada pensando en aquellas personas que han tenido muy poco encuentro con herramientas IDE (Entorno de Desarrollo Integrado) para creación de interfaces gráficas o edición avanzada de código; aunque no se quedará en una iniciación básica, sino que se abordarán algunos de los aspectos principales de esta herramienta.

Antes de empezar, es necesario aclarar bien los conceptos en cuanto a lo que se conoce como una herramienta IDE para desarrollo de software y su relación con algún lenguaje de programación como Java, .Net, php, html, etc.

Cuando un estudiante se empieza a enfrentar al mundo Java, que es el lenguaje que se ha difundido con más intensidad en la universidad, todos "aprenden" a programar en el software JCreator, y de hay en adelante todos llaman a ese software como Java. Esto es un gran error, ya que Java es un gran conjunto de librerías, módulos y Api's que entre todas conforman un lenguaje de programación de alto nivel y que pueden ser programadas con cualquier editor de texto, es decir que podemos programar Java hasta en el simple block de notas que trae todo sistema operativo; y para poder compilar y ejecutar el código que digitemos, podemos hacerlo sin necesidad de ningún editor especial; tan solo una llamada al comando de compilación desde la línea de comandos (Símbolo del sistema) es suficiente para compilar, encontrar los errores de un programa y posteriormente ejecutarlo.

Herramientas IDE, como JCreator, Netbeans, Eclipse, Sun Studio Creador, y muchas otras más, son solo editores de texto más avanzados que el Block de Notas, y que con botones sencillos dan la facilidad al usuario de compilar, revisar y corregir errores y ejecutar los programas digitados en ellos.

Así que JCreator o cualquier otro editor *NO es java*, es solo un IDE que permite escribir código fuente para muchos otros lenguajes.

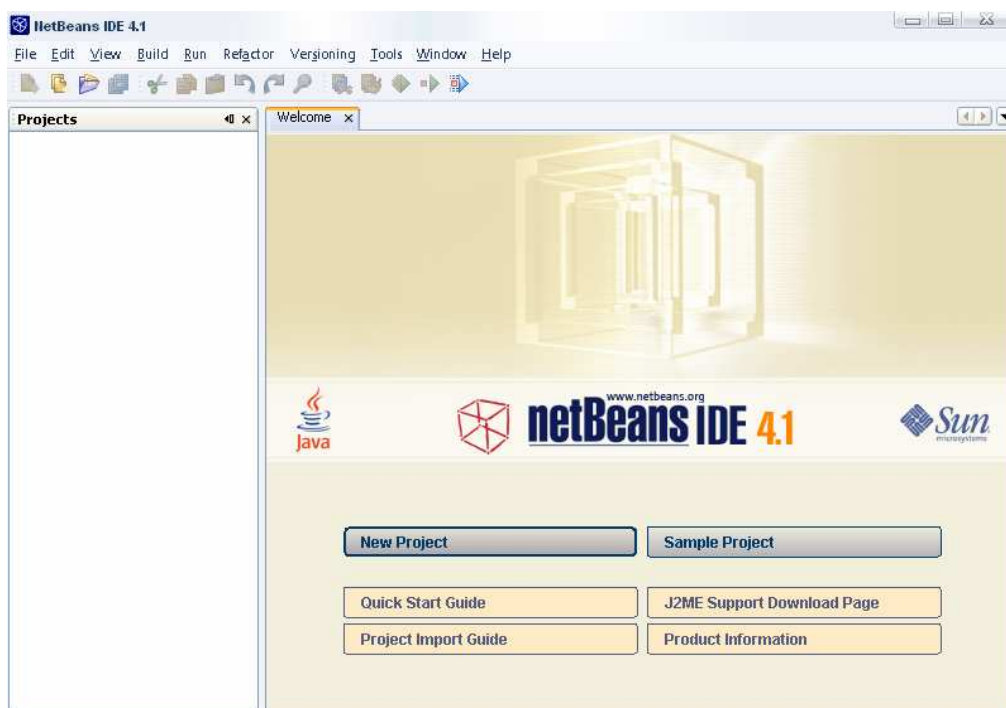
Teniendo en cuenta esta aclaración, y conociendo el propósito de este tutorial, dejo claro que no se pretende dar un curso de Java sobre Netbeans, ya que se sobreentiende que el lector ya tiene algunos conceptos básicos sobre el lenguaje.

*El desarrollo de este tutorial está basado en NetBeans v. 4.1 aunque la forma de trabajo también aplica para otras versiones superiores con solo pequeñas variaciones en el sistema gráfico, en las cuales no entraremos en detalle.*

## INICIANDO NETBEANS Y MONTANDO PROYECTOS

Para empezar a trabajar con esta herramienta, es necesario definir lo que en los IDE de este tipo se conoce como un proyecto. Un proyecto es un conjunto de archivos en los que viene incluido el código fuente, el classpath (rutas de los binarios de java y del proyecto actual), algunas instrucciones respecto a la ejecución del proyecto y algunas cosas más.

Cuando abrimos por primera vez el entorno de NetBeans (quizá se abra un diálogo de bienvenida, el cual podemos cerrar en la esquina superior izquierda de la imagen principal), el entorno tendrá una apariencia como la siguiente:



La manera más simple de crear un proyecto es utilizar la opción File | New Project (Ctrl+Shift+N) y seleccionar la primera de las opciones "Java Application".

Ahora digita el nombre del proyecto, el cual será el mismo de la carpeta en la cual se alojará todo lo que hagamos. Asegúrese de revisar la carpeta en la cual quedará almacenado el proyecto para que después recuerde en donde puede acceder a sus ficheros sin necesidad de abrir nuevamente el IDE.

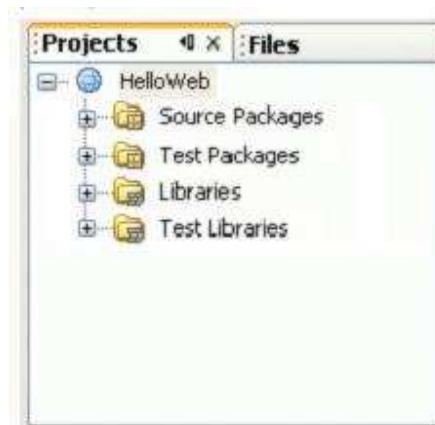
Antes de dar click en Next, Revisa las dos opciones:

*"Set as a Main Project"*

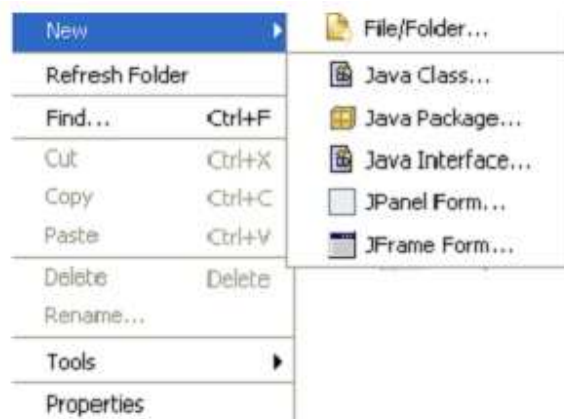
*"Create a main class".*

La primera de ellas solo déjala activa si desea que el proyecto que va a crear actualmente sea la aplicación principal dentro del IDE (es recomendado dejarlo activo). Y la segunda opción indica si el asistente creará una clase inicial la cual contendrá el método `main()` para la ejecución del proyecto (si solo se usa netbeans como IDE para la creación de interfaces gráficas, es necesario desactivarla. En caso de querer llegar a montar toda una aplicación completa, si deberá usarla).

Ahora si puede finalizar el asistente y encontrar en la parte izquierda una división algo así:

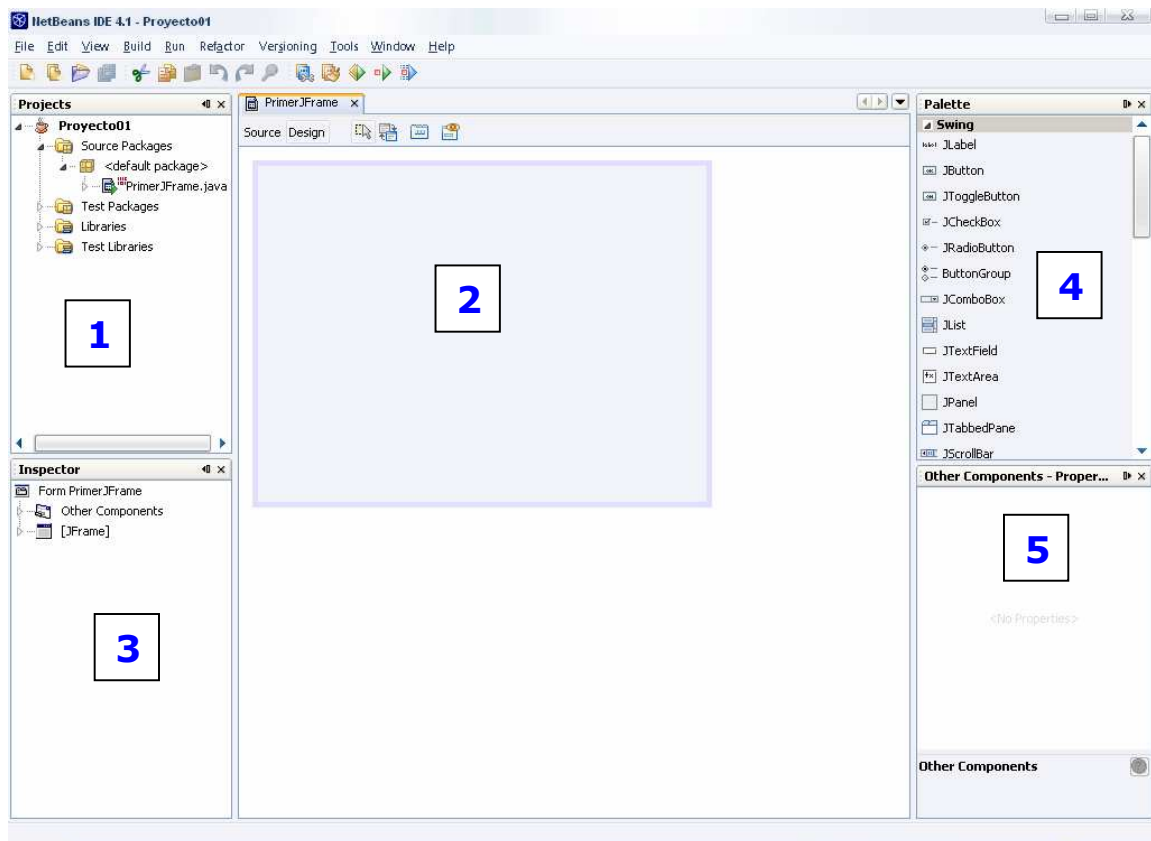


Luego de tener correctamente el proyecto montado, se puede crear todo lo que necesitamos a partir de un click derecho sobre el nombre del proyecto y escoger una de las interfaces gráficas que brinda el asistente de Netbeans:



Básicamente se trabajará con JFrame y JPanel (esto depende el tipo de aplicación a montar) aunque existen otras más tales como JDialog, JApplet JInternalFrame y otras que hacen parte del AWT pero que por ahora no trabajaremos.

Inicialmente seleccionaremos el JFrame el cual mostrará una caja de diálogo en la cual es necesario indicar el nombre de la clase que contendrá el código fuente generado por el JFrame y finalmente obtener una ventana como esta:



A partir de aquí explicaremos con detalle cada sección de toda la ventana que compone el IDE.

## Zona 1 - Projects

Como vemos, el nombre del proyecto aparece en la esquina superior y en negrita, a partir de el se ramifica el árbol de las subcarpetas, las cuales agrupan los códigos fuente de las clases junto con sus respectivos enlaces a sus interfaces gráficas (en caso de tenerlas), las librerías usadas, y algunos otros que por el nivel de este tutorial, aun no son de importancia.

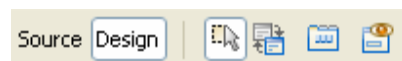
En la primera de ellas (Source Package) estará básicamente todo lo que desarrollemos, desde las clases más sencillas de solo código hasta las complejas interfaces gráficas.

Un clic derecho sobre esta carpeta source permitirá de ahora en adelante la creación de nuevas clases, ya sean gráficas o no.

## Zona 2 - Form

Esta será la zona que mostrará todo el trabajo que realicemos en el proceso de diseño de nuestra interfaz. En la parte más superior de ella, se encuentra una pestaña la cual indica el nombre de la clase gráfica y una pequeña X para poder cerrar las clases que no necesitemos.

En la barra de botones que se encuentra en esta zona tenemos:



*Source:* Activa la vista de edición de código, en la cual se puede acceder directamente a la programación pura en lenguaje java y que nos permitirá la inserción de código personalizado que no sea generado automáticamente por netbeans.

*Design:* Activa la vista de diseño gráfico del formulario (esta opción viene activa por defecto).



*Selection Mode:* Es la herramienta que permite seleccionar y manipular los objetos.



*Test Form:* Muestra una Vista Previa de la interfaz gráfica.

**Zona 3 – Inspector:** Esta zona funciona a modo de árbol; es el navegador de los objetos presentes en nuestra interfaz gráfica; es decir que allí se mostrará una lista jerárquica de todos los componentes colocados en el formulario, y desde ahí se controlará el orden de estos, sus respectivos nombres y con el clic derecho sobre cada objeto, se podrá modificar las propiedades que cada uno posee.

## Zona 4 – Palette

Aquí se encuentran los componentes gráficos que tienen las librerías Swing y AWT de Java. Con solo hacer clic sobre el objeto deseado y luego clic sobre el área de trabajo se irá diseñando la interfaz y añadiendo componentes.

## Zona 5 – Properties

Tal vez sea esta la zona más importante del entorno de programación gráfica de este IDE. En ella se encuentran las propiedades especiales y atributos que tiene cada componente gráfico. En una interfaz todo componente, desde el más externo como el JFrame hasta el más simple como un JSeparator tiene ciertas propiedades y comportamiento ante los eventos que permiten una alta personalización del diseño de los objetos colocados en el área de trabajo.

## CREACIÓN DE LA INTERFAZ

Una vez definidas las funciones de cada zona, ya podemos iniciar a crear todo el diseño de la interfaz gráfica que queremos.

Contando ya con un JFrame que hemos creado al inicio del proyecto, iniciamos por configurar la forma en la que el JFrame alojará y organizará los componentes, es decir que vamos a definir el "Administrador de Contenido":

Con *Clic derecho* sobre el JFrame, JPanel, JDialog o la interfaz que tengamos, encontramos la primera de las opciones: "*Set Layout*" y en ella usamos la opción "*Null Layout*". Con esto ya tenemos establecida la configuración que queremos.

Ahora es necesario definir el tamaño y centrado de la ventana dentro del escritorio. Empezamos por seleccionar el área de trabajo con un clic izquierdo sobre un área vacía del JFrame. Luego en la zona de Propiedades, hacemos clic sobre la tercera pestaña: "*Code*" en la cual se puede modificar la forma en la que se generará el código de la interfaz. En esta parte buscamos la Opción "*Form Size Policy*", la cual tiene por defecto "*Generate Pack()*" y ahora hay que cambiarla al valor "*Generate Resize Pack()*".

Al hacer esto, se activa la opción "*Form Size*", en la cual trae por defecto el valor "*400,300*" lo cual indica el ancho y alto inicial de la ventana respectivamente.

Seguidamente podemos regresar a las propiedades usando la primera pestaña de "*Properties*".

## AÑADIENDO OBJETOS

Como ya hemos notado, en la zona 5 (Palette) están todos los componentes que podemos usar en nuestra interfaz. Todos ellos pueden ser colocados con solo seleccionar el objeto deseado y dar click sobre la posición en donde quedará ubicado.

Al colocar cualquier objeto en la interfaz, dicho objeto aparece con un texto predeterminado, con clic derecho sobre el objeto aparece la opción "*Edit Text*" la cual permite escribir el texto deseado sobre el objeto.

Para mejorar la presentación del texto que mostremos en pantalla, algunos componentes permiten modificar el color, tipo y tamaño de letra que se muestre. Esto se hace con las propiedades *Font* y *Color* ubicadas en la zona de propiedades que aparece al seleccionar el objeto al que se le aplicarán los cambios.

Con el objeto seleccionado y arrastrando el clic sobre los bordes se modifica el tamaño; es recomendable dejar el tamaño un poco más grande de lo que se ve en Netbeans, ya que al ejecutar la aplicación como un proyecto normal, la presentación de la interfaz varía un poco.

De esta manera podemos ir

En cualquier momento podemos ver una vista previa del diseño con el botón "*Test Form*" anteriormente ya nombrado.



Para ver realmente como se verá la interfaz cuando es ejecutada como una aplicación normal, buscamos en la barra de herramientas el botón "*Run Main Project*"; o también podemos ir a la zona 1 (projects) y dar clic derecho sobre el nombre de la clase que queremos ejecutar, luego en el menú contextual, seleccionamos "*Run File*".

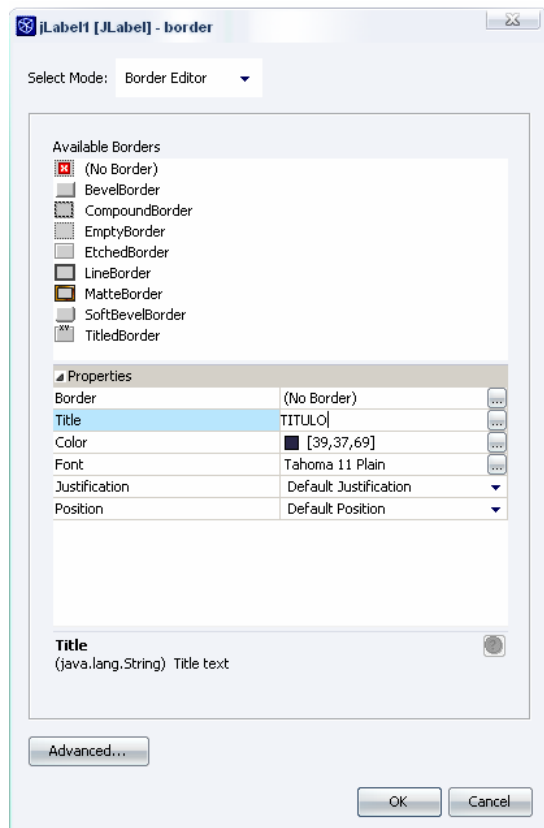
## PROPIEDADES AVANZADAS

Algunos objetos tienen propiedades únicas de cada uno de ellos lo que permite un estudio más centrado. Algunos casos son:

### JLabel y JButton

Básicamente estos componentes sirven para mostrar etiquetas con textos o el valor de una variable y para colocar botones comunes y corrientes. Pero estos componentes no solo muestran textos, también se puede mostrar imágenes como íconos, logos, fondos y demás.

Esto se logra con la opción *Icon* que al igual que muchas, se encuentra en la zona de propiedades del objeto. Con un click sobre el cuadrito con puntos suspensivos, se activa un cuadro de diálogo para escoger la ruta en donde se encuentra almacenada la imagen que se va a mostrar.



Otra opción muy interesante es el borde, que en Netbeans encontramos como "*Border*" y la cual permite escoger entre varios tipos de bordes ya predefinidos por Java y hasta colocar imágenes personalizadas como bordes.

En cuanto al JLabel, es interesante el efecto que se crea si dejamos el texto vacío y usamos el borde de tipo "Titled Border" el cual nos permite agregar texto en el borde de la etiqueta.



Este Title Border permite hacer como marcos interesantes para alojar otro tipo de componentes como se muestra en el ejemplo ubicando 2 JRadioButtons.

## JMenu

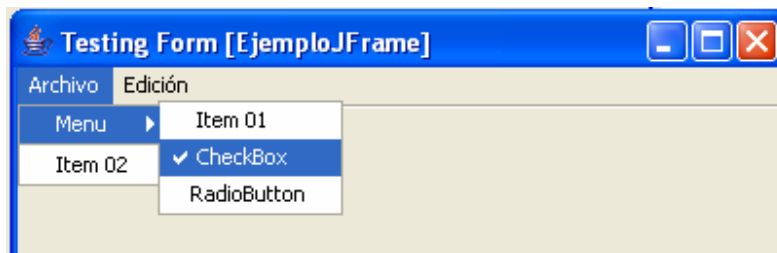
Este componente permite agregar la típica barra de menús que casi todo programa posee.

De la misma manera que lo JButton, los menús mostrados en la barra se cambian de texto con clic derecho y luego en "Edit Text".

Para agregar más menús a la barra, se da clic derecho sobre un área vacía la misma barra y al final del cuadro desplegable, clic sobre Add JMenu.

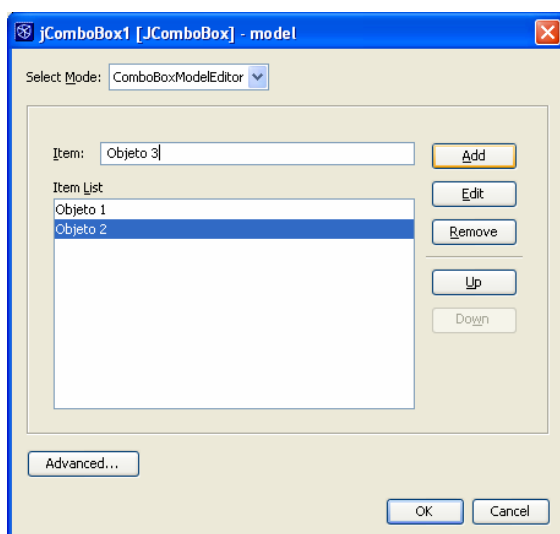
Y para agregar Ítems o contenido dentro de cada menú, le da clic derecho sobre el JMenú al que se le quiere añadir contenido y en la parte inferior de la lista, con la opción Add encontramos los componentes que se pueden agregar.

Si observa el *Inspector de Objetos*, se dará cuenta de que bajo la barra de menú van apareciendo los JMenu, JMenuItem y demás componentes que hemos añadido al menú, así que para modificar el texto que mostrará cada uno de ellos, es necesario seleccionar el objeto, y luego en la zona de propiedades, con la opción *Text*, cambiamos dicho texto.



## JComboBox

Componente que muestra una lista desplegable.



Para añadir los textos que serán mostrados en la lista, seleccionamos el JComboBox, y luego en la zona de propiedades buscamos la opción *Model* la cual muestra un cuadro de diálogo que permite ir añadiendo textos con el botón "Add", y luego de añadirlos, cambiarles de posición con los controles Up y Down, editar o eliminar su contenido.

Este JComboBox permite acceder a cada uno de los componentes listados por medio de los métodos *getSelectedIndex()* y *getSelectedItem()*; el primero retorna la posición del objeto seleccionado y el segundo retorna un object que representa la selección.



## JTable - JScrollPane

Este componente permite crear tablas de filas y columnas para desde sencillos formularios o listados hasta complejas hojas de cálculo tipo Excel.

Este componente, aunque se puede, No se debe colocar directamente en el área de trabajo, primero debe colocarse un **JScrollPane**, el cual permite la utilización de barras de desplazamiento horizontal y vertical en caso de que la cantidad de celdas sobrepase el tamaño de la vista.

Primero seleccionamos el JScrollPane y lo ubicamos en la pantalla; este aparece inicialmente como un pequeño punto, pero con un clic sobre este punto y arrastrando el Mouse en diagonal hacia arriba y la izquierda, podemos acomodar el tamaño de la vista que permitirá este scroll.

Luego seleccionamos el JTable de la Paleta y hacemos clic sobre el JScrollPane que ya hemos acomodado anteriormente.

Aparece un JTable con una cantidad de filas y columnas predeterminadas, pero estas se pueden cambiar haciendo uso de la opción *model* que se encuentra en la zona de propiedades de este objeto. Esta opción abre un cuadro de diálogo que permite agregar o quitar filas y columnas, escoger el tipo de dato que debe ir en cada celda (String, int, double, etc...) y permitir cuales celdas serán editables o no.

Select Mode: TableModelEditor

Table Model

Table Settings Default Values

Specify Title and Column Types Here:

Column	Title	Type	Editable
1	Código	Integer	<input type="checkbox"/>
2	Disponible	Boolean	<input type="checkbox"/>
3	Nombre	String	<input checked="" type="checkbox"/>
4	Valor	Double	<input checked="" type="checkbox"/>

Insert  
Delete  
Move Up  
Move Down

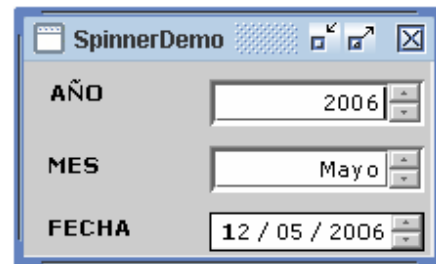
Rows: 4 + - Columns: 4 + -

Advanced...

OK Reset to default Cancel

## JSpinner

Componente que facilita el incremento o decremento de alguna variable, que por defecto es numérica, aunque también se puede usar para mostrar un rango de variables de texto, o hasta crear un objeto más complejo como un seleccionador de fechas.



En este ejemplo, el primer JSpinner es usado en conjunto con un SpinnerNumberModel, el cual soporta una secuencia de números doubles o int. Además puede especificar los valores máximos y mínimos permitidos, el valor inicial y la cantidad de cada incremento o decremento. Para definir esto, accedemos a la opción *model* de la zona de propiedades de este JSpinner, y en el cuadro de diálogo que aparece, seleccionamos la opción User Code y colocamos el siguiente código:

```
new javax.swing.SpinnerNumberModel(inicial, min, max, incremento)
```

Y lógicamente, reemplazamos las variables por el valor que deseamos.

El segundo JSpinner es usado con un SpinnerListModel el cual muestra los valores definidos en un arreglo de objetos o en una lista.

El tercer JSpinner es usado con SpinnerDateModel y un DateEditor, con lo cual el JSpinner soporta una secuencia de fechas.

Desde netbeans, estas propiedades y funcionalidades adicionales no se pueden agregar directamente por medio de la interfaz gráfica. Este tipo de personalización que soportan los spinner es solo posible editando el código y agregando estos modelos.

Este tutorial está encaminado por ahora solo a la creación de interfaces gráficas desde netbeans, así que el código fuente de personalización de este tipo de objetos se desarrollará en otro tutorial dedicado a la programación avanzada de interfaces swing.

Más info: <http://java.sun.com/docs/books/tutorial/uiswing/components/spinner.html>

Los componentes hasta ahora nombrados se han especificados un poco en detalle debido a que una buena utilización de estos, permite obtener una interfaz más agradable; la extensa lista de los demás componente de java swing no han sido detallados debido a su fácil utilización, ya que el usuario puede ir probando, colocando objetos y personalizando su interfaz a su medida.

A pesar de que Netbeans es un entorno de desarrollo muy completo y podemos editar tanto código como GUI queramos, es muy recomendable usar esta herramienta solo como editor gráfico, ya que la edición de código desde este ide no tiene un buen nivel de usabilidad que permita un buen control de código, corrección de errores y demás tareas de manera facil y agradable.

## EDICIÓN DE CÓDIGO FUENTE

Si hemos usado netbeans hasta ahora para la creación de interfaces gráficas, es como necesario preguntarnos, ¿Dónde está todo el código que genera nuestra interfaz?

Como ya he mencionado anteriormente, en la parte superior del área central de trabajo o FORM, se encuentra una pequeña barra de herramientas y el primero de los botones de esta barra es uno llamado "Source". Al hacer clic aquí pasaremos a la parte en donde como podemos ver, se encontrará todo el código de la interfaz que hemos desarrollado hasta ahora.

En el código que vemos aparecen varias secciones, unas de ellas aparecen resaltados con azul claro. Estas zonas son propias de netbeans puesto que son el enlace con la interfaz gráfica y por lo tanto no se pueden modificar manualmente

Inicialmente en esta zona se encuentran algunos comentarios sobre la fecha, nombre de clase y autor. Seguidamente está la declaración de la clase, con su respectiva herencia según la interfaz que se haya creado (Júrame, JPanel, JDialog, JApplet, etc..)

Una vez iniciada la clase, aparece el método Constructor de la clase, el cual tan solo hace una llamada al método initComponents(), y lo que este método hace es generar toda la interfaz gráfica. Un poco más debajo de este constructor hay una etiqueta con el símbolo "+" a su derecha y que dice "Generated Code", al expandir esta zona, en ella encontraremos el cuerpo del método initComponents() en el que veremos en detalle todo el código que va definiendo cada componente ubicado en la pantalla.

Si anteriormente si hemos activado la opción "generate resize code()", vamos a encontrar al final de este método dos líneas de código algo extrañas:

La primera de ellas es una llamada a la clase Dimension y se crea un objeto llamado screenSize, el cual almacena la resolución actual de la pantalla (800x600, 1024x768, etc...). Y la segunda línea establece el setBounds para la ventana gráfica. Como bien se sabe, el setBounds es el método que define la Posición y el tamaño de cualquier componente gráfico, en este caso, lo será de la ventana que hemos creado.

Los parámetros que se le pasan a este método son calculados por netbeans y lo que hacen es que siempre que se ejecute la aplicación, nuestra interfaz aparezca en el centro del monitor.

Al finalizar este método, y en caso de haber creado un JFrame, se encuentra el método main() del programa, el cual crea un hilo para luego hacer la llamada al constructor de la misma clase y hacer visible la ventana creada.

Este hilo que crea netbeans no es necesario si lo que estamos creando es una aplicación sencilla y en la cual no necesitamos manejar varios hilos. Por lo tanto toda la declaración e iniciación del hilo podemos quitarlo libremente y dejar tan solo la línea completa de la llamada al constructor de la clase junto con la inicialización de la ventana gráfica.

Por último encontramos en azul y encerrado entre comentarios, la zona de declaración de todas las variables que hacen referencia a los componentes gráficos que hemos creado.

Si te fijas bien en esas variables, aparecerán muchas con nombres extraños o con una aparente numeración, por lo tanto si vamos a digitar código manualmente nos quedará complicado saber a que componentes queremos hacer algún tipo de referencia ya que los nombres asignados no son fáciles de recordar.

Por lo tanto en este punto, es necesario volver a la zona de diseño de nuestra interfaz gráfica.

Ahora, en la zona del INSPECTOR, encontramos todos los componentes de la ventana creada y como vemos, aparecen los nombres extraños que hemos visto en la zona de código. Para corregir el problema, tan solo bastará con seleccionar cada componente, hacer clic derecho sobre el y con la opción de *Rename...* asignar un nombre acorde a lo que representa dicho objeto.

Recordemos que los nombres que le demos a estos componentes son los nombres de las variables que harán referencia a ellos, por lo tanto no podemos colocar nombre con espacios, tildes, símbolos raros ni mucho menos empezar un nombre con un número. Además, para seguir el estándar de escritura de código en los lenguajes de programación, estos nombres de variables deben iniciar con letra minúscula.

Una vez asignados los nombres correctos a los componentes de la ventana, ahora si podemos volver a la zona de *Source* ó código fuente.

Finalmente podemos pasar todo el código aquí generado a otro editor más ameno para la escritura de código como lo es JCreator ó Eclipse.

Para hacer esto, lo más fácil es seleccionar todo el código, y usar la opción de COPIAR, del menú edición (o presionar Ctrl.+V) y luego en el editor de nuestra preferencia, colocamos el texto con la opción PEGAR.

Como podemos ver muy claramente, el código creado por Netbeans es un toque extenso puesto que en cada línea a la que se hace referencia a algún objeto gráfico, siempre se está haciendo la llamada al paquete javax.swing, pero esto es fácil solucionarlo con las opciones de reemplazo de todo editor de texto, de esta manera tan solo será necesario colocar el import respectivo al inicio de la clase.

En caso de que no queramos hacer uso de otro editor, sino que queramos mantener todo el desarrollo de la aplicación desde netbeans, entonces nos ubicamos en el método constructor de la clase, y luego de la llamada al método initComponents(), procedemos a editar todo el código que la aplicación requiera.

---

Hasta aquí este tutorial pretende ser una ayuda para la iniciación en la creación de interfaces gráficas, mas no un curso completo sobre la edición avanzada de componentes gráficos. Estos temas serán tratados en otros tutoriales, puesto que como ya mencioné, dicha edición se hace es trabajando directamente con el código fuente.

La referencia completa sobre netbeans se encuentra en su sitio oficial:

<http://www.netbeans.org>

---

**TUTORIAL DESARROLLADO POR  
JULIAN ALEJANDRO LAMPREA  
UNIVERSIDAD DEL QUINDIO  
ARMENIA - 2006**