

Netbeans.

Inicio Rápido

Autor: Tomás Girardi

Fecha: 15-04-09

Versión del Documento: 1.0

NETBEANS.....	1
INICIO RÁPIDO.....	1
1. INTRODUCCIÓN.....	3
2. INSTALACIÓN.....	4
3. LA VENTANA ESTÁNDAR.....	5
4. EJEMPLO DEL DESARROLLO DE UN PROYECTO.....	7
4.1. CREACIÓN.....	7
4.2. EDITANDO EL FORM (GUI).....	9
4.3. AGREGANDO CÓDIGO.....	11
4.4. CORRIENDO EL PROGRAMA.....	11
4.5. DEBUGEANDO.....	12
5. LOS “TIPS” DEL AYUDANTE.....	14
5.1. INTRODUCCIÓN.....	14
5.2. AGREGANDO ARGUMENTOS A LA EJECUCIÓN.....	14
5.3. ¿Y QUE PASA CON LOS ARCHIVOS DE ENTRADA Y SALIDA?.....	15
5.4. CREANDO EL .JAR.....	18
5.5. “COMPLETE CODE”: POSIBLEMENTE EL TRUCO MÁS ÚTIL DE TODOS.....	18
5.6. LAS SOLUCIONES SUGERIDAS.....	20
5.7. NOTA.....	20
6. PLUGINS.....	21

1. Introducción.

Netbeans es una IDE sumamente completa, fácil de usar, cómoda y de excelente calidad; y es completamente gratis. Es muy famosa entre los programadores de java hoy en día, por lo que hay mucha información al respecto.

En este documento intentaré darle algunas de las pautas para comenzar a usar este programa.

2. Instalación

Antes de instalar NetBeans, instale el JRE y el JDK.

- JRE: https://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDSCDS_Developer-Site/en_US/-/USD/ViewProductDetail-Start?ProductRef=jre-6u13-oth-JPR@CDS-CDS_Developer
- JDK: https://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDSCDS_Developer-Site/en_US/-/USD/ViewProductDetail-Start?ProductRef=jdk-6u13-oth-JPR@CDS-CDS_Developer

Descargue netbeans desde <http://www.netbeans.org/downloads/index.html>.

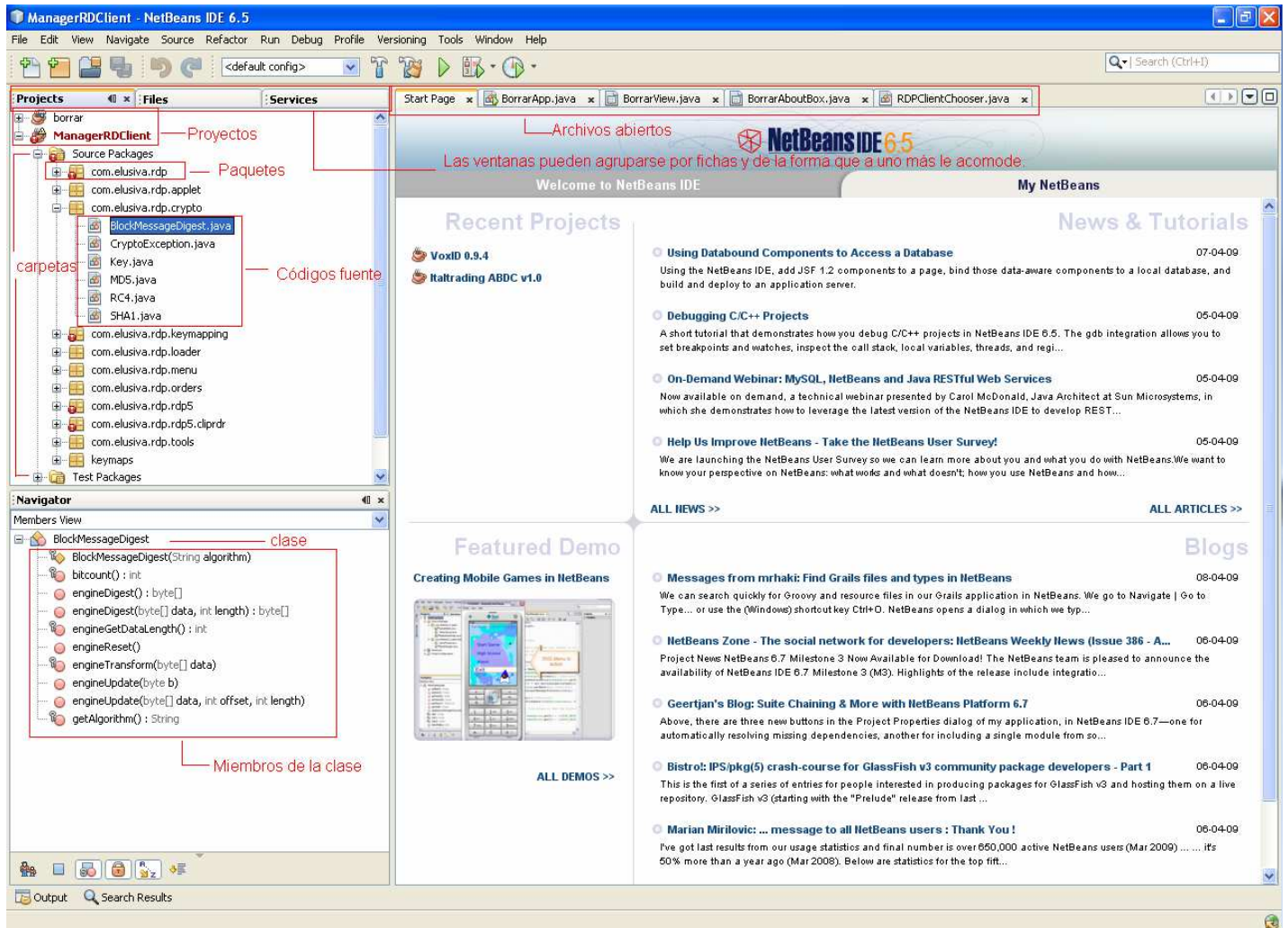
Podrá ver que se le presentan varias opciones de descarga, cada una con más o menos plugins. Le sugiero que descargue la versión más básica (Java SE) para partir. Esta versión es más liviana, se descarga más rápido y ocupa menos espacio en su disco duro, además, muchas veces las personas terminan descargando herramientas de netbeans que nunca van a utilizar. En el ramo elo329 sólo utilizará las herramientas presentes en Java SE, si más adelante quiere hacer desarrollos web en Java entonces necesitará las herramientas para java EE, si desea hacer desarrollo para dispositivos móviles necesitará las herramientas para java ME, y así sucesivamente.

Por otra parte, puede descargar las herramientas que necesite de manera sumamente simple desde netbeans y cuando realmente las necesite (ver sección “plugins”).

Ejecute el instalador. Éste intentará encontrar de manera automática la última versión de su JRE y JDK, pero puede ser que falle o simplemente le indique una versión que no es realmente la última. En estos casos, debe indicarle la ruta a la última versión de forma manual.

3. La Ventana Estándar.

Cada vez que inicie Netbeans, posiblemente vea algo así:



Projects: permite tener a mano todos los proyectos en que uno está trabajando actualmente o en los que ha trabajado en el último tiempo. A través de la expansión o colapso de los nodos en el árbol uno puede ir navegando a través de los elementos que conforman el proyecto. Sugiero hacer clic derecho sobre el proyecto y revisar la ventana de propiedades.

Files: permite navegar a través de las carpetas del proyecto. A veces algunos elementos pueden no aparecer en el árbol del proyecto en “Projects”, pero si en “Files”. Por ejemplo, los archivos de salida al ejecutar o debugear un proyecto.

Services: maneja conexiones a algunos servicios. Los más comunes son los de Bases de datos. Son una ayuda sumamente útil si uno realiza proyectos que interactúan con servidores de base de datos.

Navigator: despliega los miembros de una clase. Se sugiere revisar su funcionamiento ya que puede ser de suma utilidad. Permite desplegar o esconder los

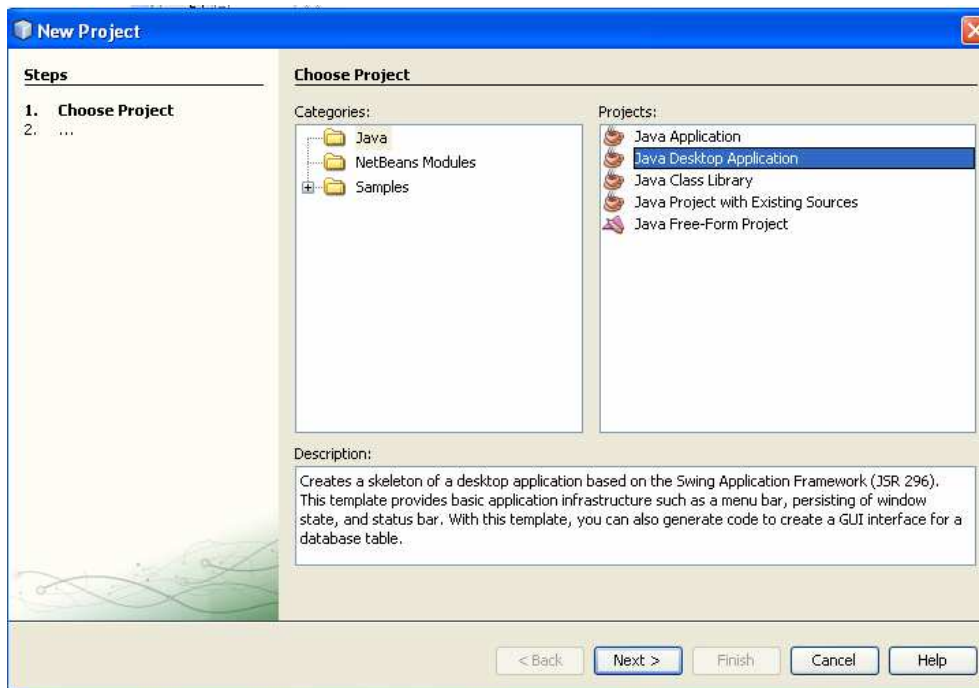
miembros heredados de una clase, desplegar o esconder los atributos de una clase, ir hasta la definición de un miembro (haciendo doble clic sobre este), encontrar ocurrencias del elemento (find usages), etc.

4. Ejemplo del Desarrollo de un Proyecto.

4.1. Creación.

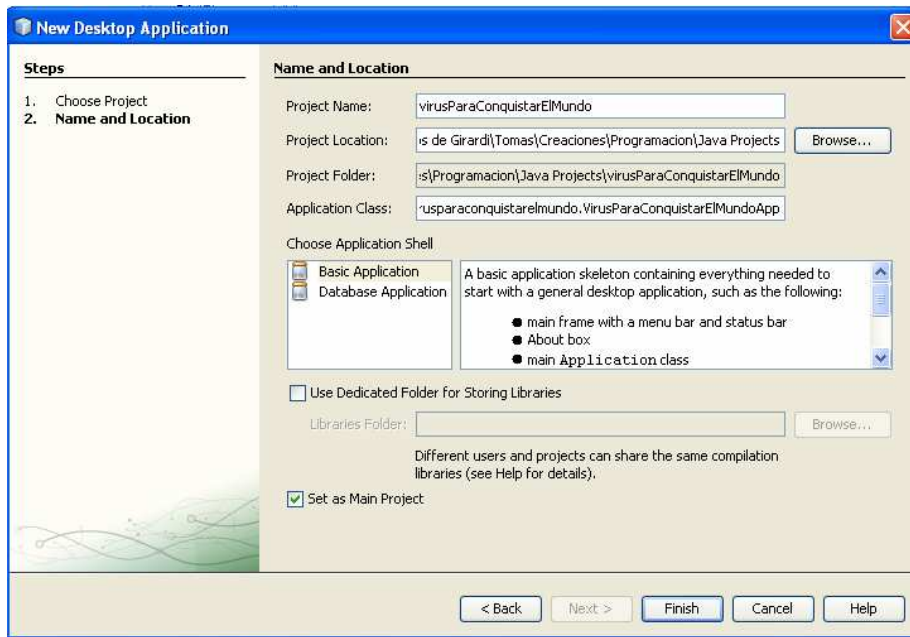
Puede presionar el botón  o el menú File -> New Project.

Elija el tipo de proyecto. Para el ejemplo elegiremos un “Java Desktop Application” (aplicación con GUI), que es más interesante a modo pedagógico que una por consola.



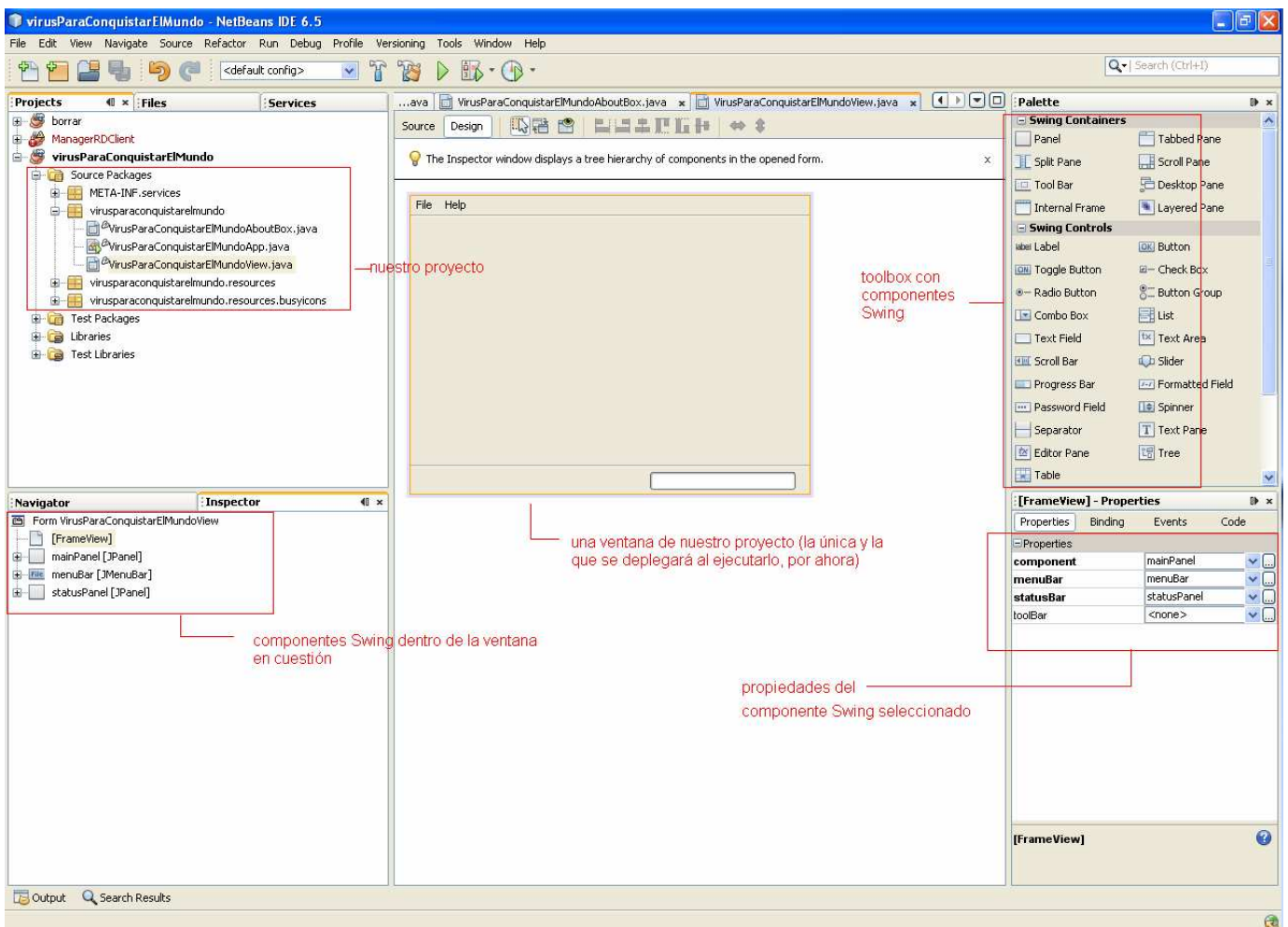
Hacemos click en Next.

Elegimos el nombre del proyecto y la carpeta donde guardarlo (intente ser ordenado). Si queremos, podemos cambiar el nombre de la clase main que se sugiere o elegir si setear como proyecto principal o no (main project, es el que se ejecutará o debugeará por defecto si uno presiona algunos botones correspondientes). Sobre el “shell”, para este ejemplo seleccionaremos una aplicación básica.



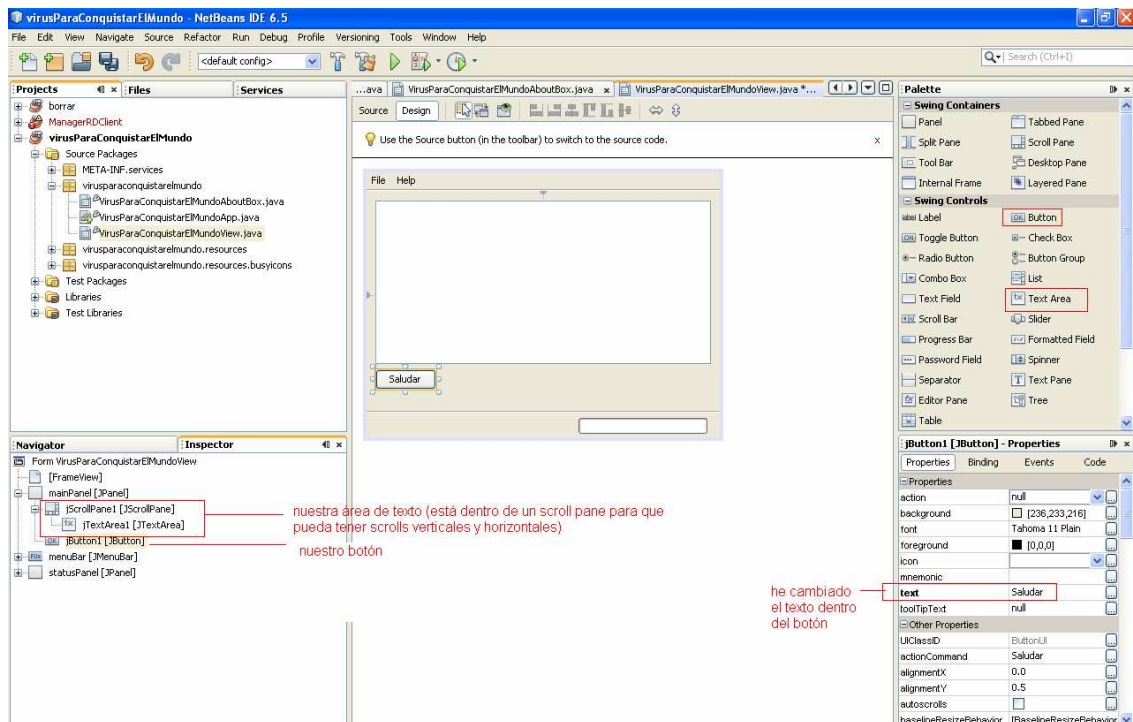
Presionamos el botón “Finish”.

Veremos algo así:



4.2. Editando el Form (GUI)

Agreguemos un área de texto y un botón.

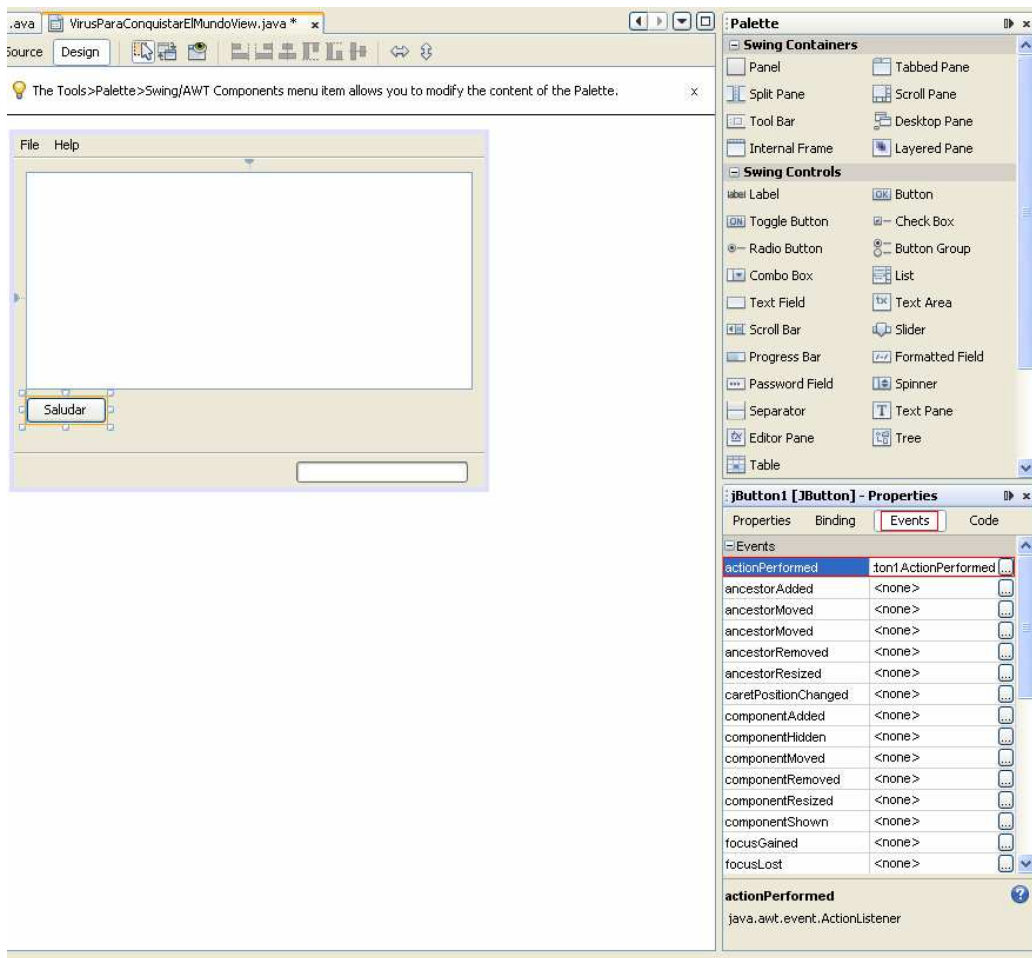


Ahora creamos una acción que se ejecutará cuando hagamos clic en el botón "Saludar".

Una forma de hacer esto es hacer clic en la tarjeta "Events" que aparece en la ventana "Properties" (En el dibujo está abajo a la izquierda).

Esta tarjeta muestra una serie de eventos que pueden ser disparados por el botón. En este caso, el que nos interesa es "actionPerformed", el cual está al inicio de la lista.

Podríamos configurar el método que se invocará cuando ocurra un evento actionPerformed disparado por el botón, pero es más fácil si simplemente hacemos doble-click sobre el campo de texto que aparece a la derecha de "actionPerformed" y que dice "<none>". Esto nos creará el método automáticamente e incluso, nos llevará a su definición una vez que el campo de texto pierda el foco ¡De seguro ya se están dando cuenta de las bondades de IDEs como esta!



```

86     if (aboutBox == null) {
87         JFrame mainFrame = VirusParaConquistarElMundoApp.getApplication().getMainFrame();
88         aboutBox = new VirusParaConquistarElMundoAboutBox(mainFrame);
89         aboutBox.setLocationRelativeTo(mainFrame);
90     }
91     VirusParaConquistarElMundoApp.getApplication().show(aboutBox);
92 }
93
94 /** This method is called from within the constructor to
95  * initialize the form.
96  * WARNING: Do NOT modify this code. The content of this method is
97  * always regenerated by the Form Editor.
98  */
99 @SuppressWarnings("unchecked")
100 Generated Code
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124 // Variables declaration - do not modify
125 private javax.swing.JButton jButton1;
126 private javax.swing.JScrollPane jScrollPane1;
127 private javax.swing.JTextArea jTextAreal;
128 private javax.swing.JPanel mainPanel;
129 private javax.swing.JMenuBar menuBar;
130 private javax.swing.JProgressBar progressBar;
131 private javax.swing.JLabel statusAnimationLabel;
132 private javax.swing.JLabel statusMessageLabel;

```

En la última imagen podemos ver la declaración del método y también están resaltados dos botones en la esquina superior izquierda: “Source” y “Design”. Estos nos permiten ver el código de la clase o la representación gráfica de su Form (como lo

veíamos antes). “Source” nos muestra el código y “Design” nos muestra la representación gráfica. Claramente sólo las clases que tienen una representación gráfica tienen estos botones.

4.3. Agregando Código.

Vamos a la definición del método que acabamos de crear (¿No sabes como ir a la definición? Recuerda el “Navigator” o también puedes ir al evento “actionPerformed” del botón en “Properties”).

```
private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {  
    // TODO add your handling code here:  
    JTextArea1.append("Hello World!\n");  
}
```

Agrega el código que aparece en la imagen.

Como podrás suponer, este código hace que cada vez que se haga click en el botón se agregue la línea “Hello World!” en el área de texto de nuestro Form.

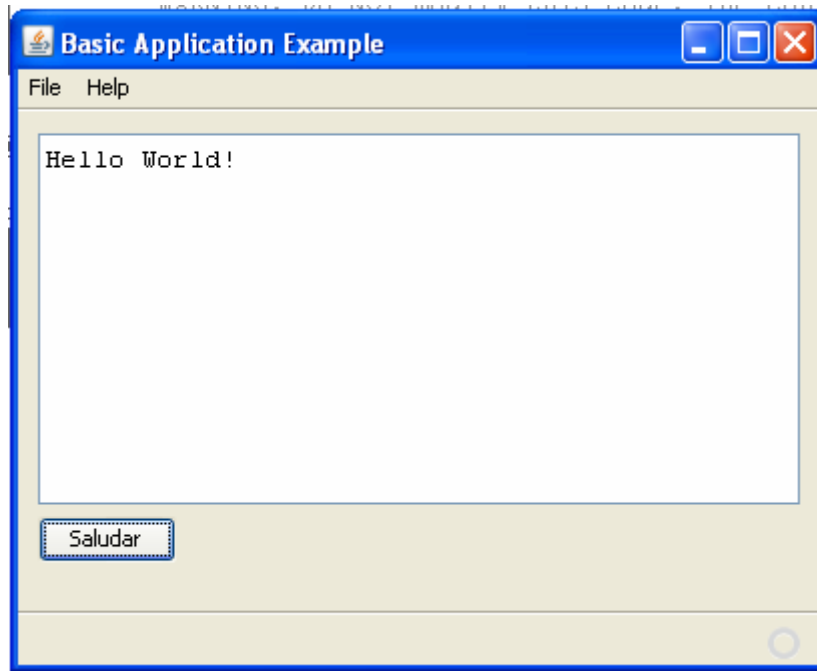
4.4. Corriendo el Programa

En este ejemplo, para correr el programa, basta con hacer clic en el botón:



Esto hará que el programa se compile (no compilará si no se han hecho cambios desde la última compilación) y lo ejecutará.

Deberíamos ver algo así al hacer click en el botón “Saludar”.



4.5. Debugueando.

Para debugear haremos algo muy similar a lo hecho al ejecutar, pero si queremos ver resultados debemos poner un punto de quiebre (breakpoint) en alguna parte del código que vaya a ser “alcanzada”.

Como ejemplo, ponga un breakpoint al inicio del código del método llamado al dispararse el evento actionPerformed del botón. Esto se hace haciendo clic una vez sobre el número de la línea donde desea poner el breakpoint. Si no ve los números, es porque tiene configurado para que estos estén ocultos. Puede buscar entre las opciones de configuración la forma de mostrarlos. En todo caso, de no estar los números de las líneas, basta que presione justo encima del borde izquierdo al lado de la línea. Debería aparecer la siguiente imagen ▾.

```

219 | private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
221 |     // TODO add your handling code here:
222 |     JTextArea1.append("Hello World!\n");
223 | }


```

La imagen ▾ aparece en casos de un breakpoint sobre la línea de declaración de un método y se “alcanzan” cuando se entra en el método. Otros breakpoints tienen la imagen ■ y la línea se cubre con rojo. Estos breakpoints se “alcanzan” cuando se está a punto de ejecutar la línea de código.

```

220 | private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
221 |     // TODO add your handling code here:
222 |     JTextArea1.append("Hello World!\n");
223 | }

```

Ahora presione el botón .

Se ejecutará la aplicación al igual que antes, pero esta vez, cuando haga click sobre el boton “Saludar” la ejecución se detendrá sobre el breakpoint.

```

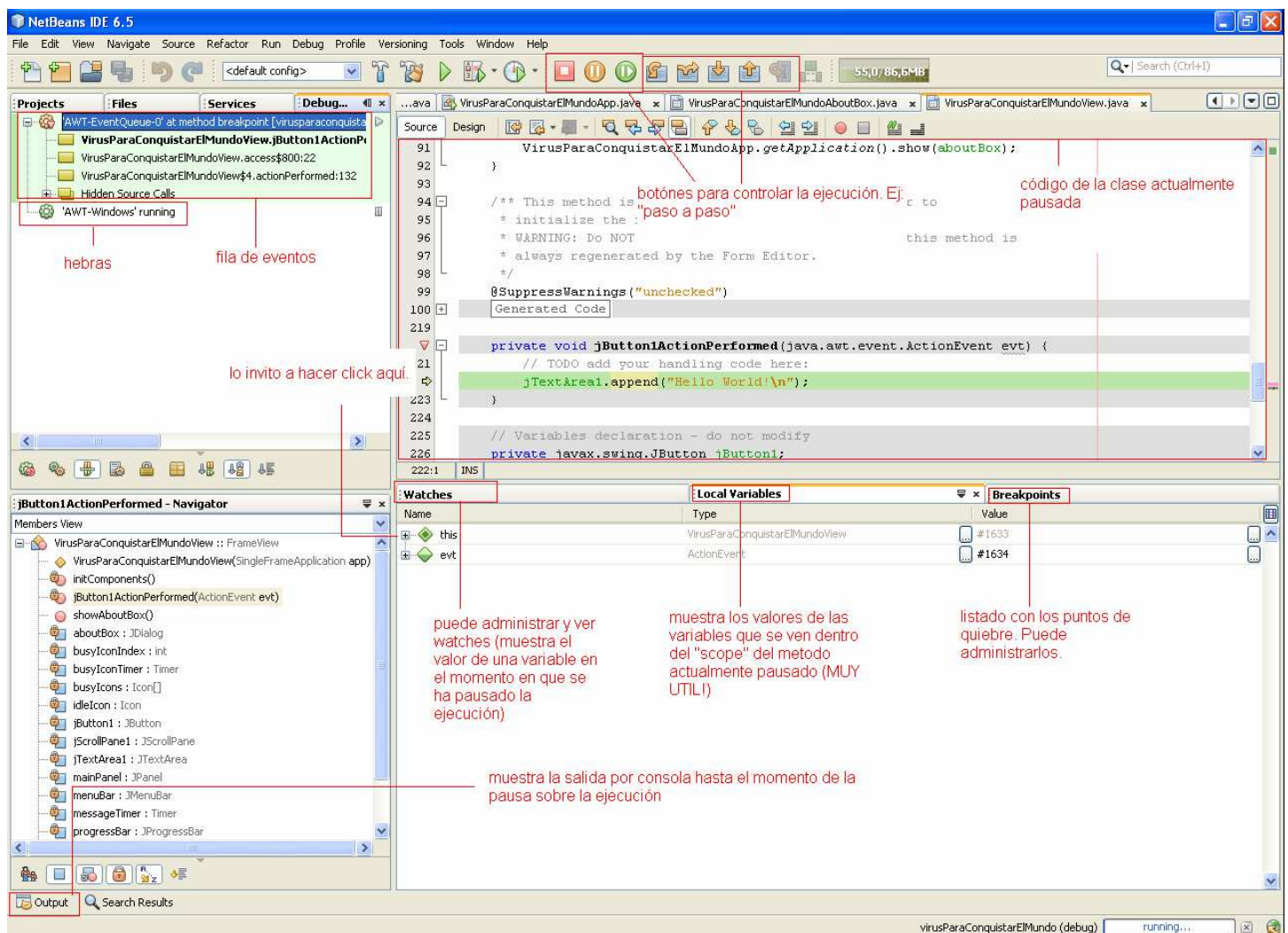
221 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
222     // TODO add your handling code here:
223     JTextArea1.append("Hello World!\n");
224 }

```

Puede ver que la línea que se ha “alcanzado” (aquella que se ejecutará a continuación) está marcada en verde.

Netbeans le permite hacer una infinidad de cosas en el modo de debugeo. Solo por citar algunas: ejecutar paso a paso, saltar hasta el cursores, poner un “watch” sobre una variable, ver la fila de eventos (Event Queue), ver las hebras, ver variables locales, etc.

Una explicación de las ventanas que posiblemente esté viendo:



Lo invito a navegar por las distintas ventanas del modo de debugeo y “jugar” un poco.

5. Los “Tips” del Ayudante.

5.1. Introducción.

A medida que uno utiliza regularmente Netbeans, podrá ir dándose cuenta de una serie de elementos que le facilitan la vida al programador.

Por desgracia, a veces puede tomarnos varios meses antes de darnos cuenta de estos elementos. Por esta razón, quise agregar algunos aquí para que se ahorren un montón de tiempo investigando o no vaya a pasar que tengan la “mala suerte” de nunca darse cuenta de un par de cosas.

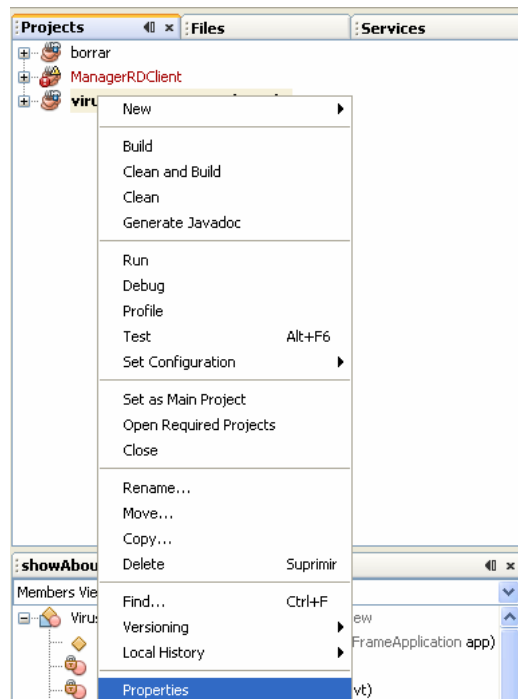
Tal vez no me acuerde de todos los tips ahora, pero si recuerdo algunos más adelante entonces los agregaré a una nueva versión del documento.

Invito a quién tenga más tips a mandarlos a la lista de correos del ramo o al correo del ayudante.

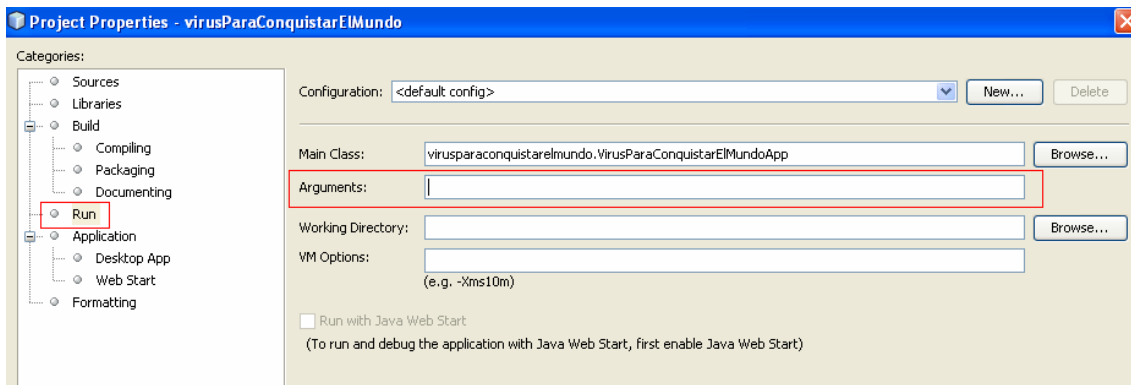
5.2. Agregando argumentos a la ejecución.

Si ha ejecutado programas en java por consola sabrá lo fácil que es darle argumentos al programa. Pero ¿Cómo lo hago si lo ejecuto desde Netbeans?

Vaya a las propiedades del proyecto (clic derecho sobre el proyecto).



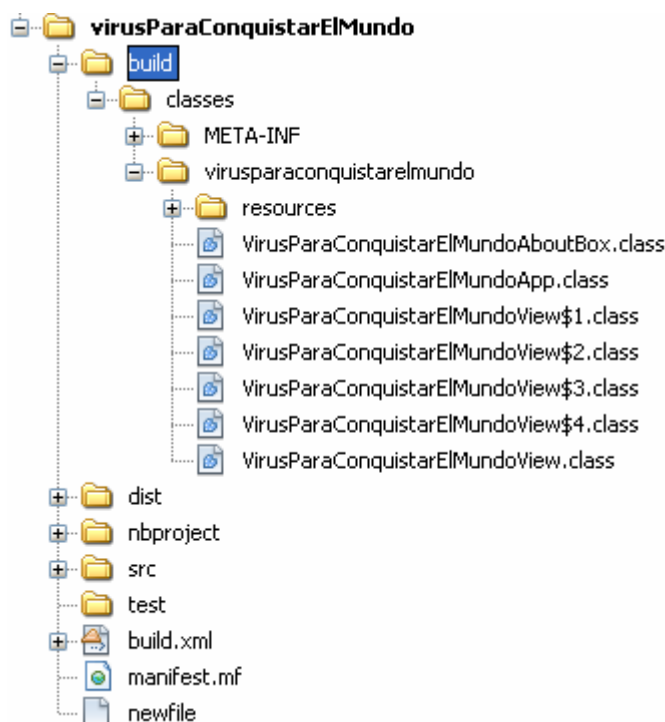
Vaya a la sección “Run” e ingrese los argumentos en el campo de texto “Arguments”.



5.3. ¿Y que pasa con los archivos de entrada y salida?

Si su programa lee o escribe datos en un archivo, es posible que tenga algunos problemas al principio. Pero si sigue estas instrucciones se solucionarán.

El directorio donde corre la aplicación por defecto es la carpeta del proyecto. Por esta razón, por defecto, la ruta de todos los archivos será relativa a la ruta hasta la carpeta del proyecto y será en esa misma carpeta donde se escriban todos los archivos de salida. Por ejemplo, si tenemos lo siguiente:

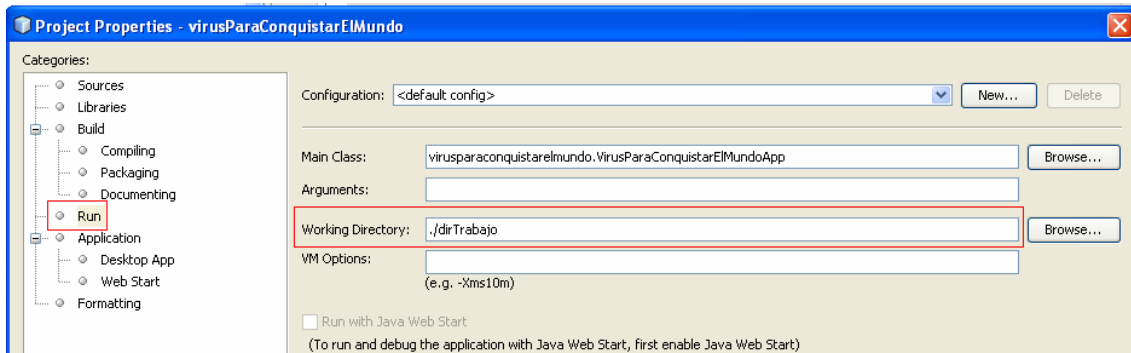


El archivo “newfile”, que está en la carpeta del proyecto (al final de la lista) será visible para el programa si lo abrimos así:

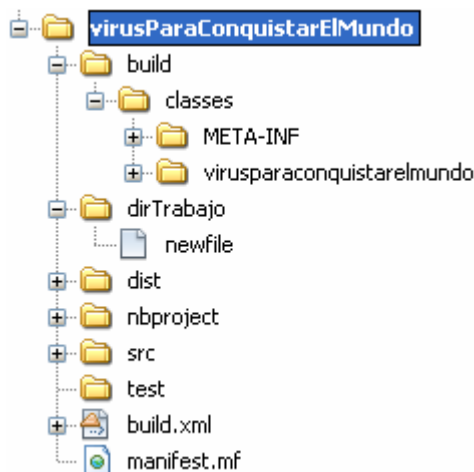
```
File archivo = new File("newfile")
```

Pero esto es bastante desordenado. Tal vez queramos que el directorio de trabajo sea otro, de forma de incluir todos los archivos de entrada y salida en una misma carpeta.

Para esto nos vamos a las propiedades del proyecto y seteamos el “Working Directory”. Aquí podemos poner una ruta relativa o absoluta, pero sugiero la primera opción, de lo contrario tendremos que cambiar este valor cada vez que movamos la carpeta o enviemos el archivo a otra persona.



En el ejemplo he creado una carpeta “dirTrabajo” donde estarán todos los archivos de entrada o salida.

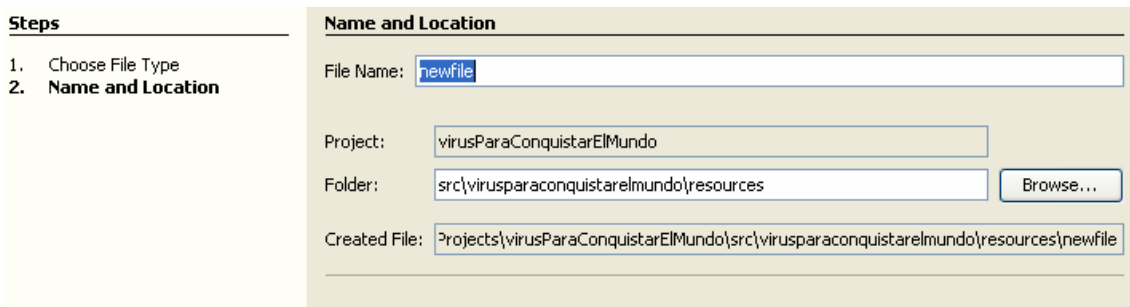
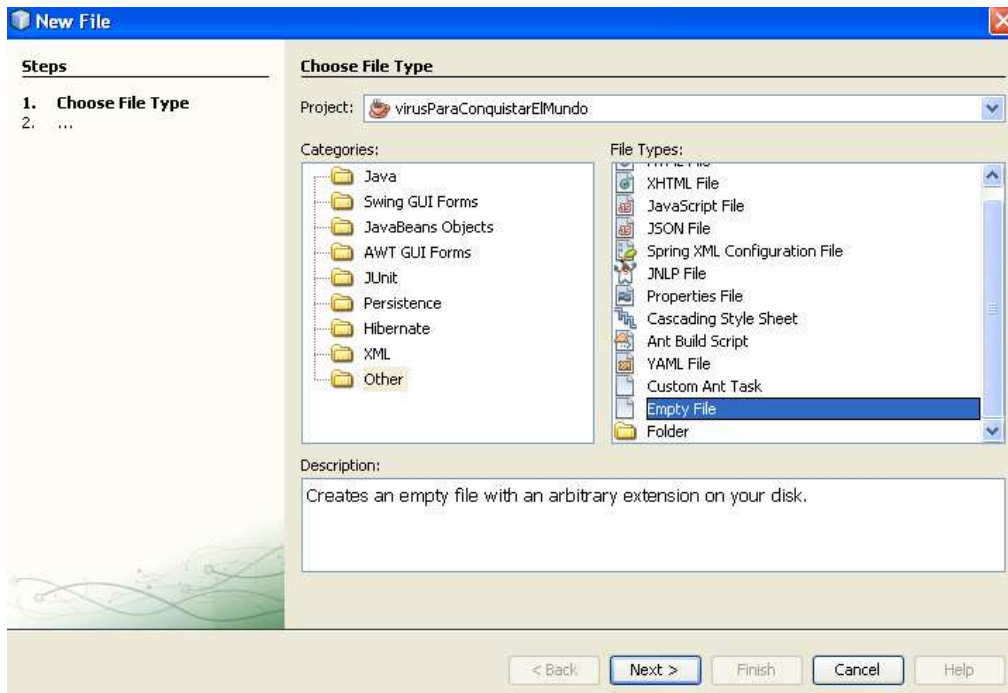
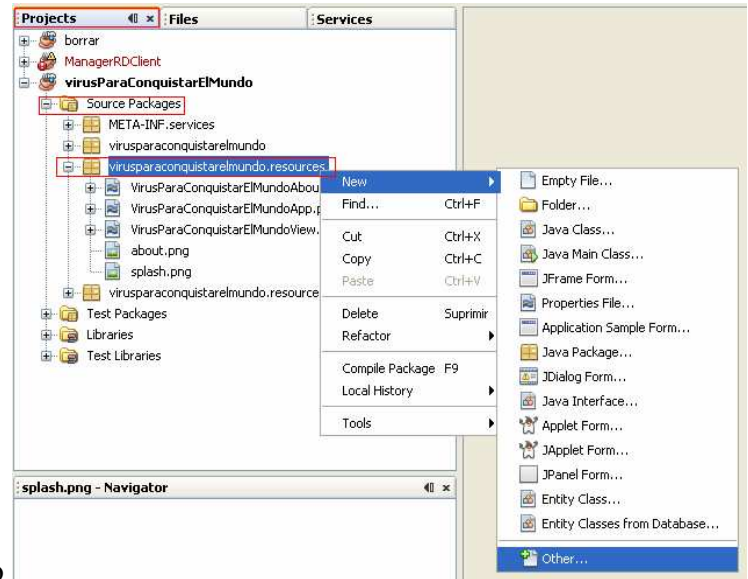
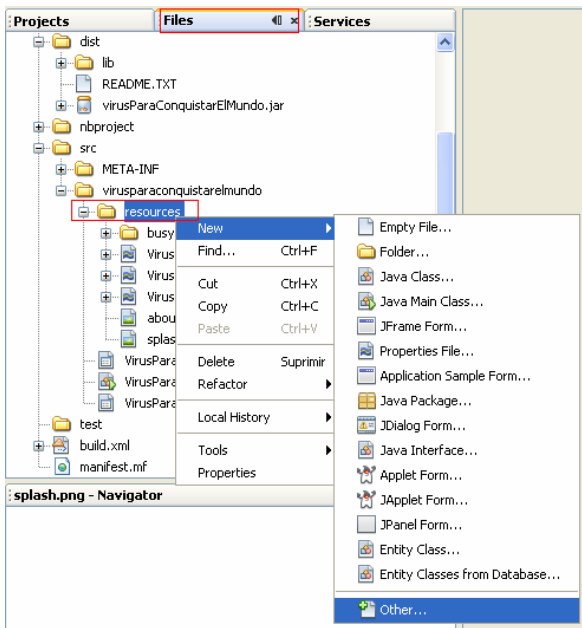


Ahora el archivo “newfile” puede seguir abriéndose como se hacía antes, pero nuestra carpeta está más ordenada.

Pero todo se puede hacer aún más ordenado. La carpeta que netbeans “entiende” como la carpeta de los recursos de entrada por defecto está en “./src/resources”. Esta puede ser la mejor opción si queremos agregar datos, imágenes, archivos xml y otros recursos para la aplicación. Generalmente se utiliza para archivos fijos, cuyo contenido no será reescrito durante la ejecución.

Hagamos un ejemplo para entenderlo.

Cree el archivo “newfile” en la carpeta src/resources.



Ahora, para abrir el archivo desde el código, tendremos que hacerlo de forma diferente. La idea es obtener una URI a través de una URL que nos da el método

getResources() de la clase a la cual pertenecen los recursos. Suena complicado ¿Cierto? En realidad no lo es, viendo el siguiente código debiese quedar bastante claro:

```
try {
    // abrimos el archivo
    File file = new File(this.getClass().getResource("./resources/newfile").toURI());
    // trabajamos con el archivo
} catch (URISyntaxException ex) {
    //TODO
}
```

Los archivos en src/resources serán copiados automáticamente en la carpeta de resources en la carpeta de la clase en “build” y también en el lugar correspondiente en el jar. En este caso no nos preocupa el directorio de trabajo, ya que usando el método getResource() siempre se encontrará el archivo si se encuentra en la carpeta resources de la clase correspondiente. Tampoco es necesario hacer cambios en el “Working Directory”.

Los archivos de salida pueden seguir siendo utilizados de la forma que se estime conveniente (sugiero la forma mencionada en el ejemplo que usa “dirTrabajo”).

Hay otras formas de manejar estos archivos y distintas “buenas prácticas”. Si alguien conoce una le agradeceré que me la describa por correo ☺.

5.4. Creando el .JAR

En netbeans es tan fácil como presionar el botón .

Se pueden hacer ajustes para personalizar un poco más el jar resultante si uno “juega” un poco con la configuración del proyecto, con el archivo MANIFEST.MF o con build.xml (contiene una serie de configuraciones sobre la compilación y creación del jar).

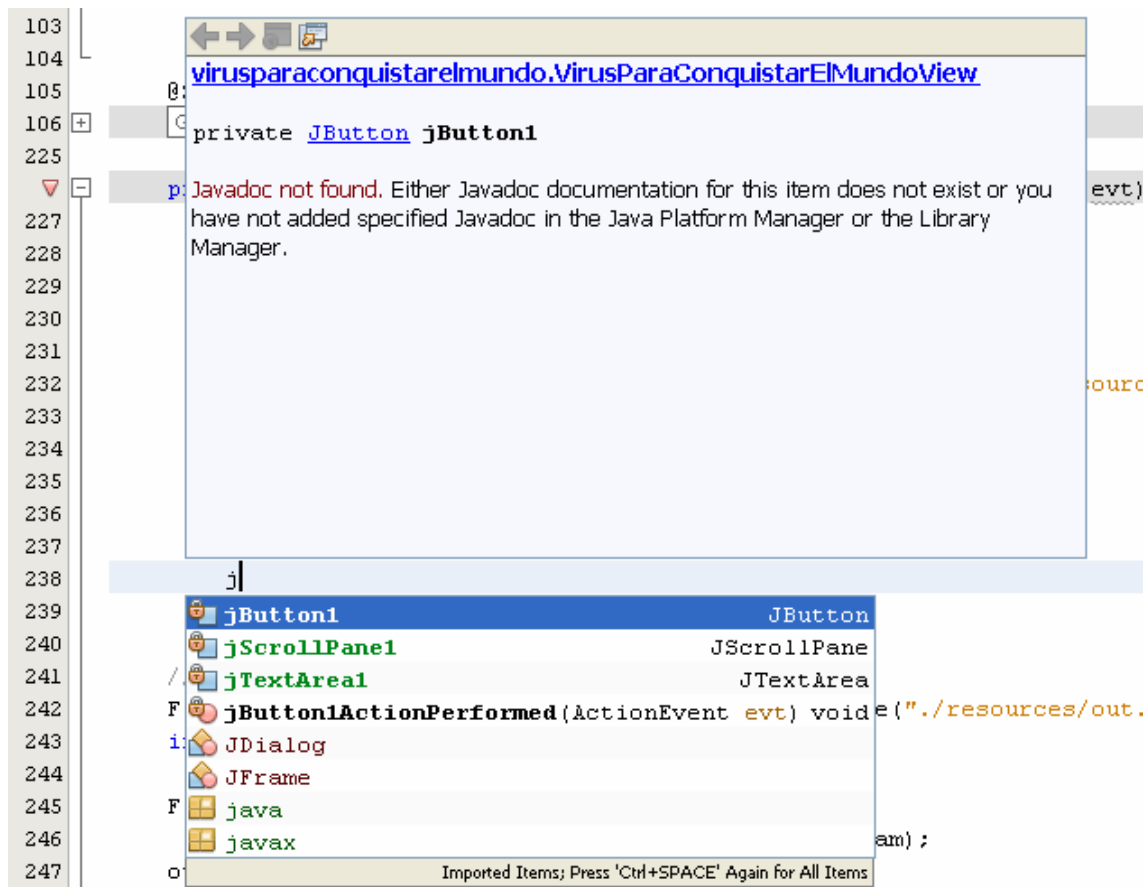
Por defecto el jar se creará en la carpeta ./dist

5.5. “Complete Code”: posiblemente el truco más útil de todos.

Mientras escribimos el código podemos desplegar un menú que nos ayudará en varios sentidos: puede mostrarnos la descripción de un método o atributo; puede mostrarnos distintas opciones de variables y métodos a utilizar; puede completarnos el nombre de un método, variable o atributo a medida que vamos escribiendo y muchos más.

Esta herramienta se llama “complete code” y la forma más cómoda de invocarla es apretando ctrl.+Espacio mientras escribimos un código. OJO: a veces hay problemas con esta combinación porque algunos hotkeys del sistema operativo o de otro programa lo tienen reservado. Un caso bien típico es Messenger, que suele emplear esa combinación para esconderse (creo que le llaman “boss alert” o algo así).

Por ejemplo. Si quiero modificar el texto dentro del botón “Saludar”, pero no me acuerdo bien del nombre, sólo se que parte con “j...”. Entonces puedo presionar Ctr+espacio y luego la tecla “j” y veré algo así



Cuando les aparezca “javadoc not found” es posible que se trate de un elemento propio que ustedes no han documentado o porque no han instalado el javadoc del JDK que están utilizando. No me extenderé sobre este tema ahora, pero es posible encontrar documentación al respecto en google.

Cuando hayan seleccionado lo que quieren escribir, basta apretar el enter y netbeans escribirá todo lo que falte.

De seguro ya les está entusiasmando la idea.

Otro caso de uso es cuando ya tenemos escrito todo aquello del método o variable que lo identifica unívocamente. En esos casos, al apretar ctrl. + espacio, netbeans rellena automáticamente el resto del nombre (no es necesario seleccionar del menú).

Por ejemplo: si escribo “jTe” (en el ejemplo que hemos desarrollado) y luego presiono ctrl.+Espacio, netbeans escribe automáticamente “jTextArea1”


También podemos usarlo cuando no recordamos bien los métodos de una clase o incluso cuando queremos buscar un método en particular y creemos que con el nombre y la descripción del javadoc podremos determinar si es lo que andamos buscando o no.

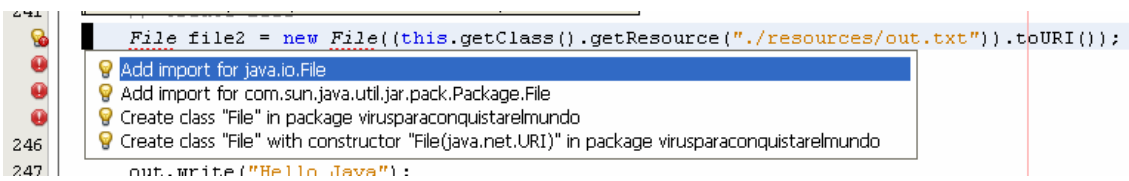
5.6. Las soluciones sugeridas

Netbeans es capaz de sugerir distintas alternativas de solución a errores y advertencias.

Por ejemplo: ¡Es capaz de agregar los bloques try-catch por nosotros cuando lo hemos olvidado!

Vemos otro ejemplo:

Al olvidar importar `java.io.File` me aparece un error y al lado izquierdo un ícono: . Si presiono sobre él, me sugiere varias soluciones. Basta que seleccione “Add import java.io.File” para que netbeans ponga la sentencia import donde corresponde.



Si nos valemos de esto, en la mayoría de los casos no haremos nosotros los imports, sino que confiaremos en netbeans para que los haga por nosotros.

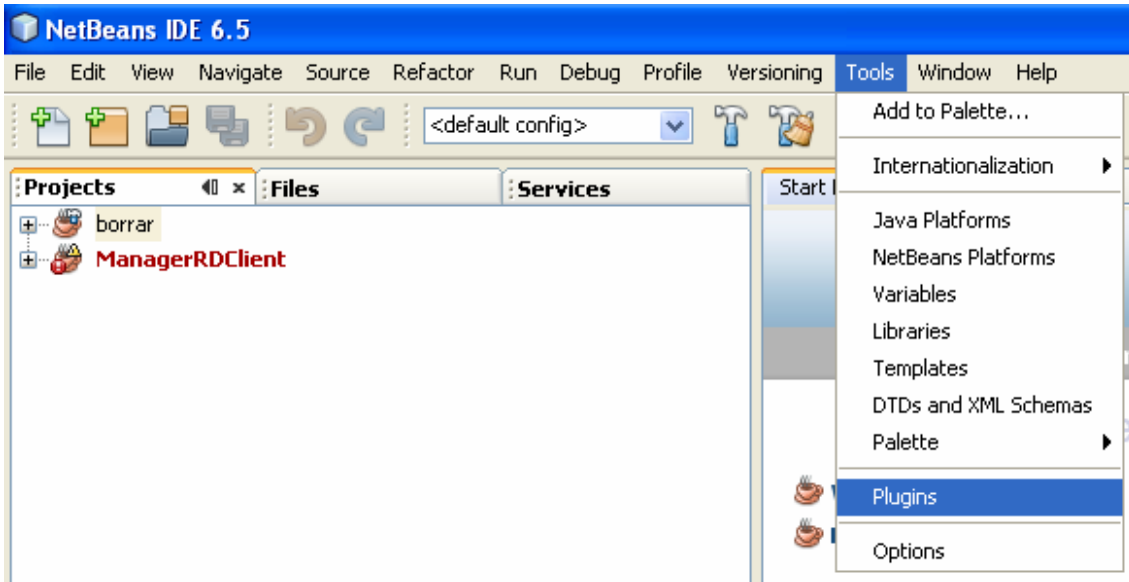
5.7. Nota

Posiblemente se me quedan varios en el tintero. Por ejemplo: como cambiar el JDK que se usa para el proyecto, como agregar librerías y JDKs, como instalar javadocs para que netbeans los use, etc. Pero no creo que sea buena idea extenderme tanto.

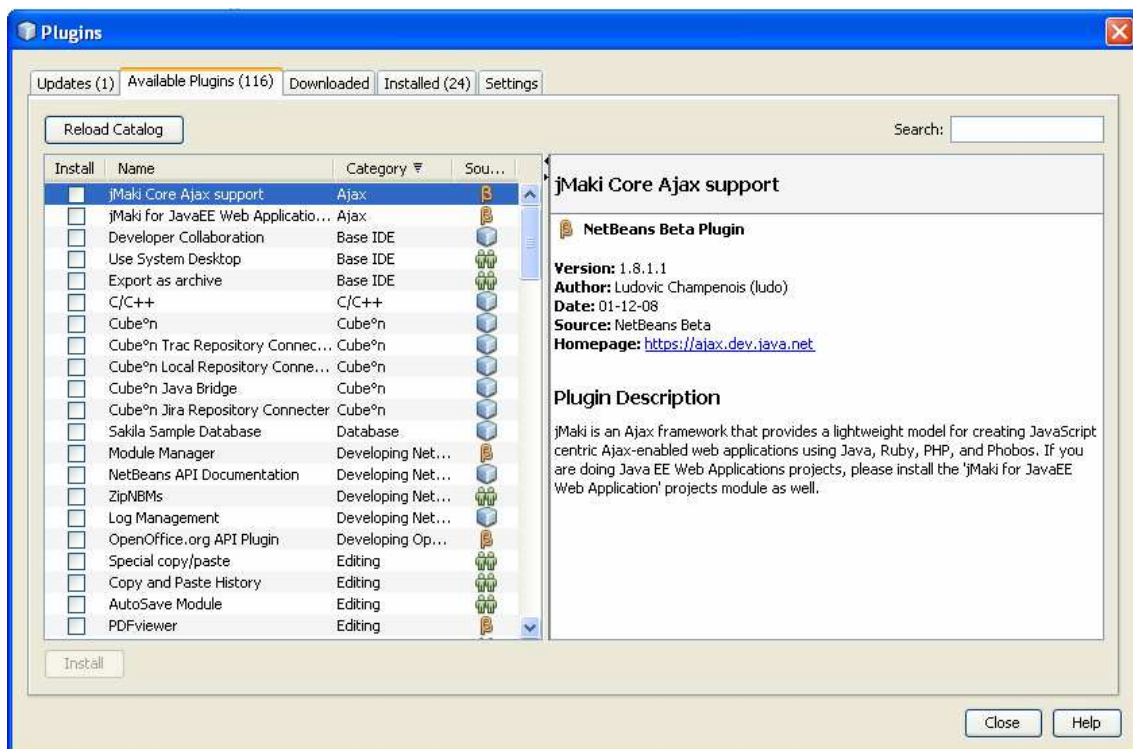
Jueguen un poco con el programa y si tienen dificultades me contactan.

6. Plugins

Puede instalar y desinstalar herramientas adicionales en netbeans en cualquier momento que lo desee. Para esto vaya a Tools -> Plugins.



Vea las opciones que tiene en “Available Plugins”



Seleccione las que desee y apriete en el botón “Install”. Estas se descargarán e instalarán automáticamente.

Una herramienta interesante es UML (para trabajar con diagramas UML), y para programadores más avanzados hay herramientas para XML y esquemas, para WSDL, layouts gráficos de JSF, etc.

Puede desinstalar las que ya no ocupe en la pestaña “Installed” y puede actualizar su versión de netbeans en “Updates”.