



**ESCUELA POLITÉCNICA DEL EJÉRCITO
SEDE LATACUNGA**

**CARRERA DE TECNOLOGÍA DE SISTEMAS E
INFORMÁTICA**

**PROYECTO DE GRADUACIÓN PREVIO A LA
OBTENCIÓN DEL TÍTULO DE TECNÓLOGO EN
COMPUTACIÓN**

TEMA:

**ELABORACIÓN DE UN MANUAL DE LA
PLATAFORMA NETBEANS IDE PARA LA
DISICOM**

ELABORADO POR:

**WALTER CELIANO TITUAÑA CUMBAL
EDWIN JESÚS TORRES CAÑIZARES**

LATACUNGA – ECUADOR

MARZO 2009

ÍNDICE

<u>I.</u>	<u>CAPÍTULO</u>	<u>- 9 -</u>
1.1	LA PLATAFORMA DE NETBEANS.....	- 9 -
1.2	HISTORIA DE NETBEANS	- 11 -
1.3	DEFINICIÓN DE NETBEANS.....	- 14 -
1.3.1	¿QUÉ ES NETBEANS?.....	- 14 -
1.3.2	PROCESO DE LIBERACIÓN DE VERSIONES DE NETBEANS	- 15 -
1.3.3	VERSIONES	- 17 -
1.3.4	RADIOGRAFÍA	- 18 -
1.3.5	ESTADO ACTUAL DE NETBEANS IDE 6.5	- 19 -
<u>II.</u>	<u>CAPÍTULO</u>	<u>- 20 -</u>
2.1	REQUERIMIENTOS E INSTALACIÓN DE NETBEANS IDE	- 20 -
2.1.1	REQUERIMIENTOS DEL SISTEMA.....	- 20 -
2.1.2	OPCIONES DE DESCARGA DEL INSTALADOR	- 20 -
2.1.3	CONFIGURACIONES RECOMENDADAS DE HARDWARE:	- 22 -
2.1.4	CONFIGURACIONES MÍNIMAS DE HARDWARE:	- 22 -
2.2	INSTALACIÓN DE NETBEANS IDE	- 23 -
2.2.1	INSTALACIÓN EN WINDOWS.....	- 24 -
2.2.2	INSTALACIÓN EN LINUX UBUNTU 8.1.....	- 28 -
2.2.3	INSTALACIÓN EN SISTEMAS OPERATIVOS SOLARIS	- 33 -
<u>III.</u>	<u>CAPÍTULO</u>	<u>- 34 -</u>
3.1	ESTUDIO Y REVISIÓN DE NETBEANS	- 34 -
3.1.1.	INTRODUCCIÓN.....	- 34 -
3.1.1.1	Conceptos Básicos del IDE	- 34 -
3.1.1.2	La Ventana Files	- 34 -
3.1.1.3	Los proyectos en el IDE.....	- 35 -
3.1.1.4	Ventana Servicios.....	- 37 -
3.1.1.5	Los Recursos de Runtime	- 37 -
3.2	CONOCIMIENTO BÁSICO DE LAS HERRAMIENTAS	- 38 -
3.2.1.	TIPOS DE DATOS	- 38 -
3.2.2.	OPERADORES.....	- 39 -
3.2.3.	ESTRUCTURAS DE CONTROL.....	- 40 -
3.2.3.1.	Sentencia if.....	- 40 -
3.2.3.2.	Sentencia WHILE	- 41 -
3.2.3.3.	Sentencia do/while	- 43 -
3.2.3.4.	Sentencia For.....	- 44 -
3.2.3.5.	Sentencia SWITCH	- 45 -

3.3	VENTAJAS Y DESVENTAJAS DE NETBEANS.....	46 -
3.3.1	Ventajas.....	46 -
3.3.2	Desventajas.....	46 -
3.4	PLANIFICACIÓN Y ELABORACIÓN DEL MANUAL DE NETBEANS.....	47 -
3.4.1	MENÚ FILE.....	47 -
3.4.1.1	Creación de un Proyecto.....	48 -
3.4.1.1.1	Java (Figura 3.4).....	50 -
3.4.1.1.2	Java Web.....	51 -
3.4.1.1.3	Java EE.....	52 -
3.4.1.1.4	Java ME.....	54 -
	Groovy.....	56 -
	NetBeans Modules.....	56 -
3.4.1.2	Name and Location.....	57 -
3.4.1.3	Creación de un archivo.....	60 -
3.4.1.4	Crear un nuevo grupo de proyecto.....	76 -
	Propiedades del Proyecto.....	78 -
3.4.2	MENÚ EDIT.....	79 -
3.4.2.1	Comando Undo (Ctrl-Z).....	79 -
3.4.2.2	Comando Redo (Ctrl-Y).....	79 -
3.4.2.3	Comando Cut (Ctrl-X).....	79 -
3.4.2.4	Comando Copy (Ctrl-C).....	80 -
3.4.2.5	Comando Paste(Ctrl-V).....	80 -
3.4.2.6	Comando Paste Formatted (Ctrl-Mayúsculas-V).....	80 -
3.4.2.7	Comando Delete.....	80 -
3.4.2.8	Comando Select All (Ctrl-A).....	80 -
3.4.2.9	Comando Find Selection (Ctrl-F3).....	80 -
3.4.2.10	Comando Find Next (F3).....	81 -
3.4.2.11	Comando Find Previous (Mayúsculas-F3).....	81 -
3.4.2.12	Comando Find (Ctrl-F).....	81 -
3.4.2.13	Comando Replace (Ctrl-H).....	81 -
3.4.2.14	Comando Find Usages (Alt-F7).....	81 -
3.4.2.15	Comando Find in Project (Ctrl-Mayúsculas-F).....	82 -
3.4.2.16	Comando Replace in Projects (Ctrl-Mayúsculas-H).....	82 -
3.4.2.17	Comando Start Macro Recording.....	82 -
3.4.2.18	Comando Stop Macro Recording.....	82 -
3.4.3	MENÚ VIEW.....	83 -
3.4.3.1	Comando Code Folds (Pliegues del código).....	83 -
3.4.3.1.1	Collapse Fold (Ctrl-Numpad - -).....	83 -
3.4.3.1.2	Expand Fold (Ctrl-Numpad - +).....	83 -
3.4.3.1.3	Collapse All (Ctrl-cambio-Numpad - -).....	83 -
3.4.3.1.4	Expand All (Ctrl-cambio-Numpad - +).....	83 -
3.4.3.1.5	Full Screen (Alt-Mayúsculas-Enter).....	83 -
3.4.3.2	Toolbars.....	84 -
3.4.3.2.1	Para agregar un toolbar a la ventana principal:.....	84 -
3.4.3.2.2	El IDE agrega un toolbar vacío a la ventana principal.....	84 -
3.4.3.2.3	Para mover un toolbar:.....	84 -
3.4.4	MENÚ NAVIGATE.....	85 -

3.4.4.1	Comando Go to File (Alt-Mayúsculas-O).....	- 85 -
3.4.4.2	Comando Go to Type (Ctrl-O)	- 86 -
3.4.4.3	Comando Go to Test (Ctrl-cambio-T)	- 86 -
3.4.4.4	Comando Go to Previous Document	- 86 -
3.4.4.5	Comando Go to Source (Ctrl-Mayúsculas-B)	- 86 -
3.4.4.6	Comando Go to Declaration (Ctrl-B).....	- 86 -
3.4.4.7	Comando Go to Super Implementation (Ctrl-Mayúsculas-P)	- 86 -
3.4.4.8	Comando Last Edit Location (Ctrl-Q).....	- 87 -
3.4.4.9	Comando Back (Alt-Left).....	- 87 -
3.4.4.10	Comando Forward (Alt-Right).....	- 87 -
3.4.4.11	Comando Go to Line (Ctrl-G).....	- 87 -
3.4.4.12	Comando Toggle Bookmark (Ctrl-Mayúsculas-M)	- 87 -
3.4.4.13	Comando <i>Next Bookmark</i> (Ctrl-Mayúsculas-Period)	- 87 -
3.4.4.14	<i>Previous Bookmark</i> (Ctrl-Mayúsculas-Comma).....	- 87 -
3.4.4.15	Next Error (Ctrl-Period).....	- 88 -
3.4.4.16	Previous Error (Ctrl-Comma).....	- 88 -
3.4.4.17	Select in Project (Ctrl-Mayúsculas-1)	- 88 -
3.4.4.18	Select in Files (Ctrl-Mayúsculas-2)	- 88 -
3.4.4.19	Select in Favorites (Ctrl-Mayúsculas-3).....	- 88 -
3.4.5	MENÚ SOURCE	- 89 -
3.4.5.1	Format (Alt-Mayúsculas-F).....	- 89 -
3.4.5.2	Shift Left (Alt-Mayúsculas-Left)	- 89 -
3.4.5.3	Shift Right (Alt-Mayúsculas-Right)	- 89 -
3.4.5.4	<i>Move Up</i> (Alt-Mayúsculas-Up).....	- 89 -
3.4.5.5	<i>Move Down</i> (Alt-Mayúsculas-Down)	- 90 -
3.4.5.6	<i>Duplicate Up</i> (Ctrl-Mayúsculas-Up)	- 90 -
3.4.5.7	<i>Duplicate Down</i> (Ctrl-Mayúsculas-Down).....	- 90 -
3.4.5.8	<i>Toggle Comment</i> (Ctrl-Mayúsculas-C)	- 90 -
3.4.5.9	Complete Code (Ctrl-Space).....	- 90 -
3.4.5.10	Insert Code (Alt-Insert)	- 90 -
3.4.5.11	Fix Code (Alt-Enter).....	- 90 -
3.4.5.12	Fix Imports (Ctrl-Shift-I)	- 91 -
3.4.5.13	Show Method Parameters (Ctrl-P).....	- 91 -
3.4.5.14	Show Documentation (Ctrl-Shift-Space)	- 91 -
3.4.5.15	Insert Next Matching Word (Ctrl-Shift-K)	- 91 -
3.4.5.16	Insert Previous Matching Word (Ctrl-K).....	- 91 -
3.4.6	MENÚ RUN	- 92 -
3.4.6.1	Run Main Project (F6)	- 92 -
3.4.6.2	Debug Main Project (Ctrl-F5)	- 92 -
3.4.6.3	Test Project (Alt-F6)	- 92 -
3.4.6.4	Run File (Shift-F6).....	- 93 -
3.4.6.5	<i>Debug File</i> (Ctrl-Shift-F5).....	- 93 -
3.4.6.6	Test File (Ctrl-F6).....	- 93 -
3.4.6.7	Debug Test for File (Ctrl-Shift-F6)	- 93 -
3.4.6.8	Finish Debugger Session (Shift-F5).....	- 93 -
3.4.6.9	Continue (F5)	- 93 -
3.4.6.10	Step Over (F8)	- 94 -

3.4.6.11	Step Over Expression (Shift-F8)	- 94 -
3.4.6.12	Step Into (F7)	- 94 -
3.4.6.13	Step Out (Ctrl-F7)	- 94 -
3.4.6.14	Run to Cursor (F4)	- 94 -
3.4.6.15	Run into Method (Shift-F7)	- 94 -
3.4.6.16	Make Callee Current (Ctrl-Alt-Up).....	- 95 -
3.4.6.17	Make Caller Current (Ctrl-Alt-Down)	- 95 -
3.4.6.18	Toggle Line Breakpoint (Ctrl-F8)	- 95 -
3.4.6.19	New Breakpoint (Ctrl-Shift-F8).....	- 95 -
3.4.6.20	New Watch (Ctrl-Shift-F7).....	- 95 -
3.4.6.21	Evaluate Expression (Ctrl-F9)	- 95 -
3.4.7	MENÚ DEBUG.....	- 96 -
3.4.7.1	Comando Debug > Debug Main Project ( Ctrl + F5)	- 97 -
3.4.7.2	Comando  Attach Debugger	- 97 -
3.4.7.3	Comando Step Into (F7 o )	- 97 -
3.4.7.4	Comando Finish Debugger Session (Shift+F5).....	- 98 -
3.4.7.5	Comandos Pause y Continuar	- 98 -
3.4.7.5.1	Comando Pause .- Detiene la Sesión	- 98 -
3.4.7.5.2	Comando Continuar.- Reanuda la Sesión	- 98 -
3.4.7.6	Comandos	- 98 -
3.4.7.7	 Step Over (F8)	- 99 -
3.4.7.8	Step Over Expression (Shift+F8).....	- 99 -
3.4.7.8.1	Step Into Next Method(Shift+F8).....	- 99 -
3.4.7.8.2	Step Out (Ctrl+F7)	- 100 -
3.4.7.8.3	Comando Run to Cursor (F4 o ).....	- 100 -
3.4.7.8.4	Debugging an Individual Project	- 100 -
3.4.8	MENÚ PROFILE	- 101 -
3.4.8.1	Comando Profile > Profile Main Project (Perfil> Perfil del Proyecto Principal) ...	- 102 -
3.4.8.2	Comando Profile > Attach Profiler (Perfil> Adjuntar Perfil)	- 102 -
3.4.8.3	Comando Profile > Profile Other > " <i>projectname</i> ".....	- 102 -
3.4.8.3.1	Para el perfil principal del proyecto	- 103 -
3.4.8.3.2	Para el perfil de un proyecto individual	- 103 -
	Para el perfil de una clase específica:	- 104 -
3.4.9	Menú Versioning	- 104 -
3.4.9.1	CVS: Versioning Options Window	- 105 -
3.4.9.2	Mercurial Global Options.....	- 105 -
3.4.9.3	Subversion: Options Window	- 106 -
3.4.9.4	Local History.....	- 106 -
3.4.10	MENÚ TOOLS	- 106 -
3.4.10.1	Comando Analyze Javadoc.....	- 107 -
3.4.10.2	Comando Apply Diff Patch	- 107 -
3.4.10.3	Comando Internationalization	- 107 -
3.4.10.4	Comando NetBeans Platform	- 108 -
3.4.10.5	Comando Variables	- 108 -
3.4.10.6	Comando Templates (Plantillas)	- 108 -

3.4.10.6.1	Las plantillas normales.....	- 109 -
3.4.10.6.2	Las plantillas de Formulario libre.....	- 109 -
3.4.10.7	Comando Palette (Paleta).....	- 109 -
3.4.10.8	Comando Plugins (otra vez).....	- 109 -
3.4.10.9	Comando Options (Opciones).....	- 110 -
3.4.11	MENÚ WINDOW.....	- 111 -
3.5	PRESENTACIONES DE EJEMPLOS EN LA PLATAFORMA.....	- 114 -
3.5.1	Ejemplo1: INCREMENTO Y DECREMENTO.....	- 114 -
3.5.2	Ejemplo2: OPERADORES LÓGICOS.....	- 115 -
3.5.3	Ejemplo3: "ARRAY".....	- 117 -
3.5.4	Ejemplo4: SENTENCIA "IF".....	- 118 -
3.5.5	Ejemplo 5: SENTENCIA "SWITCH".....	- 119 -
3.5.6	Ejemplo6: SENTENCIA "WHILE".....	- 121 -
3.5.7	Ejemplo7 : SENTENCIA "DO WHILE".....	- 122 -
3.5.8	Ejemplo8 : SENTENCIA "FOR".....	- 123 -
3.6	DISEÑO DE APLICACIÓN WEB.....	- 124 -
3.6.1	Aplicaciones Web en NetBeans.....	- 124 -
3.6.1.1	Configurando un Proyecto de Aplicación Web.....	- 124 -
3.6.2	Creando y Editando los Archivos Fuente de la Aplicación Web.....	- 128 -
	Creando un Paquete Java y un Archivo Fuente Java.....	- 128 -
3.6.3	Generando Métodos Getter y Setter.....	- 130 -
3.6.4	Editando el Archivo JSP (JavaServer Pages).....	- 132 -
3.6.5	Creando un Archivo JSP (JavaServer Pages).....	- 134 -
3.6.6	Compilando y Ejecutando un Proyecto de Aplicación Web.....	- 137 -
3.7	APLICACIÓN WEB MEDIANTE DISEÑO GRÁFICO.....	- 139 -
3.7.1	Creando y Editando los Archivos Fuente de la Aplicación Web.....	- 142 -
<u>IV.</u>	<u>CAPITULO.....</u>	<u>- 143 -</u>
4.1	CONCLUSIONES Y RECOMENDACIONES.....	- 143 -
4.1.1	CONCLUSIONES.....	- 143 -
4.1.2	RECOMENDACIONES.....	- 143 -
4.2	BIBLIOGRAFÍA.....	- 144 -

TABLA DE FIGURAS

Figura. 1. Configurando el instalador.....	- 24 -
Figura. 2. Bienvenido al Instalador de NetBeans en Windows.....	- 24 -
Figura. 3. Acuerdo de Licenciamiento en Windows	- 25 -
Figura. 4. Seleccionar el JDK junto con el NetBeans IDE.....	- 26 -
Figura. 5. Resumen para instalación de NetBeans IDE en Windows.....	- 26 -
Figura. 6. Instalación de NetBeans IDE en Windows.....	- 27 -
Figura. 7. Instalación completa de NetBeans IDE en Windows	- 27 -
Figura. 8. Bienvenido al instalador de NetBeans IDE en Linux (Ubuntu)	- 28 -
Figura. 9. Acuerdo de Licenciamiento para Linux (Ubuntu)	- 29 -
Figura. 10. Especificar donde instalar NetBeans IDE en Linux (Ubuntu)	- 29 -
Figura. 11. Especificar servidores y versión de JDK para Linux (Ubuntu).....	- 30 -
Figura. 12. Especificar carpeta para Glassfish	- 30 -
Figura. 13. Resumen para instalación de NetBeans en Linux (Ubuntu).....	- 31 -
Figura. 14. Instalación de NetBeans IDE en Linux (Ubuntu).....	- 31 -
Figura. 15. Instalación completa de NetBeans IDE en Linux (Ubuntu).....	- 32 -
Figura. 16. Pantalla principal de NetBeans IDE 6.5.....	- 32 -

Figura 3. 1. Menú FILE.....	- 47 -
Figura 3. 2. Pantalla principal de NetBeans 6.5.....	- 48 -
Figura 3. 3. Icono de New Project	- 49 -
Figura 3. 4. Ventana New Project.....	- 49 -
Figura 3. 5. Ventana New Project Java Web	- 51 -
Figura 3. 6. Ventana New Project Java EE.....	- 52 -
Figura 3. 7. Ventana New Project Java ME	- 54 -
Figura 3. 8. Ventana New Project NetBeans Modules	- 56 -
Figura 3. 9. Ventana New Java Application	- 57 -
Figura 3. 10. Creación Aplicación Hola Mundo.....	- 58 -
Figura 3. 11. Ventana New File	- 60 -
Figura 3. 12. Ventana New File Java Class	- 60 -
Figura 3. 13. Ventana New File Swing GUI Form	- 63 -
Figura 3. 14. Ventana New File JavaBeans Objects.....	- 65 -
Figura 3. 15. Ventana New File AWT GUI Forms	- 66 -
Figura 3. 16. Ventana New File JUnit	- 67 -
Figura 3. 17. Ventana New File Persistence.....	- 68 -
Figura 3. 18. Ventana New File Hibernate	- 70 -
Figura 3. 19. Ventana New File XML	- 71 -
Figura 3. 20. New file de Otros Tipos de aplicaciones	- 73 -
Figura 3. 21. Ventana New Java Class.....	- 75 -
Figura 3. 22. Ventana de la aplicación Hola Mundo.....	- 75 -
Figura 3. 23. Ventana Project Group	- 76 -
Figura 3. 24. Ventana Create New Group	- 77 -
Figura 3. 25. Ventana MENÚ EDIT	- 79 -
Figura 3. 26. Ventana MENÚ VIEW	- 83 -
Figura 3. 27. Ventana NAVIGATE	- 85 -
Figura 3. 28. Ventana MENÚ SOURCE	- 89 -
Figura 3. 29. Ventana MENÚ RUN.....	- 92 -
Figura 3. 30. Ventana MENÚ DEBUG.....	- 96 -
Figura 3. 31. Ventana MENÚ DEBUG Pause.....	- 98 -

Figura 3. 32. Ventana MENÚ DEBUG Continue	- 98 -
Figura 3. 33. Ventana de Comandos.....	- 98 -
Figura 3. 34. Ventana MENÚ PROFILE	- 101 -
Figura 3. 35. Ventana MENÚ VERSIONING	- 104 -
Figura 3. 36. Ventana MENÚ TOOLS	- 106 -
Figura 3. 37. Cuadro de Opciones de Comando	- 110 -
Figura 3. 38. Ventana MENÚ WINDOW	- 111 -

Figura 3.5. 1. Ventana Proyecto Nuevo Web Application	- 124 -
Figura 3.5. 2. Ventana Nueva Web Application	- 125 -
Figura 3.5. 3. Ventana Nuevo Web Application Server and Setting	- 126 -
Figura 3.5. 4. Ventana Nuevo Web Application Frameworks.....	- 126 -
Figura 3.5. 5. Ventana Proyecto Hola Web	- 127 -
Figura 3.5. 6. Ventana Nuevo Class Java	- 129 -
Figura 3.5. 7. Ventana Nuevo Class Java Nombre y Ubicación.....	- 129 -
Figura 3.5. 8. Ventana Encapsular campos.....	- 130 -
Figura 3.5. 9. Ventana Hola_Web Palette	- 132 -
Figura 3.5. 10. Ventana Nuevo JSP File.....	- 135 -
Figura 3.5. 11. Ventana Insert Use Bean.....	- 135 -
Figura 3.5. 12. Ventana Insert Bean Property.....	- 136 -

Figura 3.7. 1 New Project Java Web	- 139 -
Figura 3.7. 2 Ventana New Web Application	- 139 -
Figura 3.7. 3 Ventana New Web Application Server and Settings	- 140 -
Figura 3.7. 4 Ventana New Web Application Frameworks.....	- 140 -
Figura 3.7. 5 Ventana Proyecto Web Diseño_Web.....	- 141 -
Figura 3.7. 6 Aplicación Web Hola Mundo.....	- 142 -

INTRODUCCIÓN

El presente documento constituye un tutorial de la plataforma de desarrollo de sistemas software denominado NetBeans IDE.

El tutorial consiste los procesos de:

- Instalación en los diferentes sistemas operativos
- Configuración y usabilidad
- Entorno de desarrollo y la interfaz gráfica

Lo ideal es que este material sea utilizado por los desarrolladores estudiantes y educadores

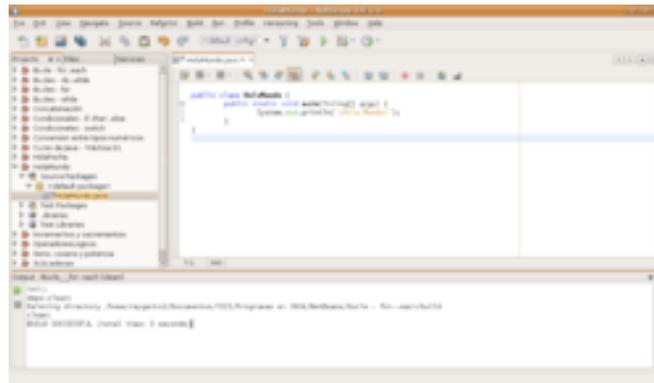
NetBeans es un entorno de desarrollo integrado (IDE) que permite editar programas en java, compilarlos, ejecutarlos, depurarlos y construir rápidamente el interfaz gráfico de una aplicación eligiendo los componentes de una paleta, etc. NetBeans es también una **plataforma de ejecución** de aplicaciones, es decir, facilita la escritura de aplicaciones Java, proporcionando una serie de *servicios* comunes, que a su vez están disponibles a través del IDE.

El Entorno de Desarrollo Integrado (IDE) **NetBeans** es un entorno de programación para varios lenguajes, incluyendo Java y C++. Este entorno de desarrollo integrado es de fuente abierta es decir, se proporciona el código fuente del entorno para que se pueda modificar de acuerdo a ciertos parámetros de licencia.

La idea de esta guía es que el alumno aprenda por si solo las principales utilidades que proporciona **NetBeans IDE 6.5** para construir la interfaz gráfico de una aplicación.

I. CAPÍTULO

1.1 LA PLATAFORMA DE NETBEANS



El **IDE NETBEANS** es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NETBEANS. El IDE NETBEANS es un producto libre y gratuito sin restricciones de uso.

El **NETBEANS IDE** es de código abierto escrito completamente en Java usando la plataforma NETBEANS. El NETBEANS IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

NETBEANS ofrece a los desarrolladores una serie de funcionalidades para los lenguajes de programación dinámicos Ruby y JavaScript, así como soporte para el último Java Standards. También nuevas mejoras en NETBEANS GUI Builder que incluyen soporte total para las nuevas tecnologías de sobremesa, como Beans Binding (JSR 295) y el entorno Swing Application Framework (JSR 296), mejor edición del código, capacidades de navegación e inspección, historia local,

soporte integrado para subversión, y mayores capacidades de personalización integradas en la distribución estándar. NETBEANS de hecho, está incluido con el soporte para NetBeans Ruby, permitiendo a los desarrolladores utilizar Ruby on Rails con el código Java existente.

De otro lado, hay que incidir que en NETBEANS 6.0 la serie completa de paquetes, como Visual Web Pack, se proporcionará a través de un único programa de instalación que permite configurar más rápidamente el entorno de desarrollo y que ya no requiere la instalación de múltiples paquetes para añadir capacidades al IDE.

Estas nuevas características de NETBEANS 6.0 se añaden a las funcionalidades de anteriores versiones de NETBEANS, que ofrecen a los desarrolladores un soporte completo para construir aplicaciones basadas en Java Platform Standard Edition (Java SE), Java Platform Enterprise Edition (Java EE), y Java Platform Micro Edition (Java ME). A este respecto, Jeet Kaul, vicepresidente de Programas para Desarrolladores de Sun, comenta que "durante los últimos dos años, hemos observado un creciente interés que ha hecho que la adopción de NETBEANS se duplicara cada año. Con NETBEANS 6.0 esperamos seguir impulsando esta gran acogida. Esto es otro ejemplo del enorme dinamismo que está experimentando.

Sun con sus iniciativas de código abierto. Con las nuevas capacidades incorporadas y la nueva experiencia de instalación, NETBEANS 6.0 se convierte en la mejor opción IDE".

También ofrece una serie de servicios de soporte para el software NETBEANS, que abarcan desde una única incidencia a completos planes de desarrollo. Además, los desarrolladores pueden mejorar sus habilidades a través de un amplio portafolio de ofertas de formación y certificación, y sacar así el máximo partido de estas innovadoras tecnologías.

1.2 HISTORIA DE NETBEANS

La historia detrás de NetBeans es interesante como sucede con cualquier pieza de software, o un trabajo de historia del arte, no la primavera plenamente formada a partir de la frente de nadie, pero está vinculada a los sueños, sudor y trabajo duro de un número de personas.

NETBEANS es una plataforma para el desarrollo de aplicaciones usando JAVA y a un entorno de desarrollo integrado (IDE). Ese (IDE) comenzó como un proyecto estudiantil en La República Checa que originalmente ellos llamaron XELFI, en 1996 bajo la tutoría de la Facultad de Matemáticas y Física en la Universidad de Charles en Praga. La meta era escribir un entorno de desarrollo integrado (IDE) para JAVA parecida a la de DELPHI. XELFI fue el primer entorno de desarrollo integrado escrito en JAVA, con su primer pre-release en 1997.

XELFI fue un proyecto divertido para trabajar, ya que las IDEs escritas en JAVA eran un territorio desconocido en esa época. El proyecto atrajo suficiente interés por lo que los estudiantes, después de graduarse, decidieron que lo podían convertir en un proyecto comercial. Prestando espacios web de amigos y familiares, formaron una compañía alrededor de esto. Casi todos ellos siguen trabajando en NETBEANS.

Tiempo después ellos fueron contactados por ROMAN STANEK, un empresario que ya había estado relacionado con varias iniciativas en la República Checa.

Él estaba buscando una buena idea en que invertir, y encontró en XELFI una buena oportunidad. Ellos se reunieron, y el negocio surgió.

El plan original era desarrollar unos componentes JAVABEANS para redes. Jarda Tulach, quien diseñó la arquitectura básica de la IDE, surgió con la idea de llamarlo NETBEANS, con el fin de describir lo que ellos harían. Cuando las especificaciones de los ENTERPRISE JAVABEANS salieron, ellos decidieron trabajar con este estándar, ya que no tenía sentido competir con él, sin embargo el nombre de NETBEANS se quedó.

En la primavera de 1999, NETBEANS DEVELOPERX2 fue lanzado, soportado Swing. Las mejoras de rendimiento que llegaron con el JDK 1.3, lanzado en otoño de 1999, hicieron a NETBEANS una alternativa realmente viable para el desarrollo de herramientas. En el verano de 1999, el equipo trabajó duro para rediseñar a DeveloperX2 en un NETBEANS más modular, lo que lo convirtió en la base de NETBEANS hoy en día.

Algo más paso en el verano de 1999. Sun Microsystems quería una mejor herramienta de desarrollo de Java, y comenzó a estar interesado en NETBEANS. En otoño de 1999, con la nueva generación de NETBEANS en Beta, el acuerdo fue realizado.

Sun adquirió otra compañía de herramientas al mismo tiempo, Forté, y decidió renombrar NETBEANS a Forté for Java. El nombre de NETBEANS desapareció de vista por un tiempo.

Seis meses después, se tomó la decisión de hacer a NETBEANS open source. Mientras que Sun había contribuido considerablemente con líneas de código en varios proyectos de código abierto a través de los años, NETBEANS se convirtió en el primer proyecto de código abierto patrocinado por ellos. En Junio del 2000 NetBeans.org fue lanzado.

NetBeans Hoy

Un proyecto de código abierto no es nada más ni nada menos que un proceso. Toma tiempo encontrar el equilibrio. El primer año, fue crucial como inicio. Los dos años siguientes, se orientó hacia código abierto. Como muestra de lo abierto que era, en los primeros dos años había más debate que implementación.

Con NetBeans 3.5 se mejoró enormemente en desempeño, y con la llegada de NetBeans 3.6, se reimplementó el sistema de ventanas y la hoja de propiedades, y se limpió enormemente la interfaz. NetBeans 4.0 fue un gran cambio en cuanto a la forma de funcionar del IDE, con nuevos sistemas de proyectos, con el cambio no solo de la experiencia de usuario, sino del reemplazo de muchas piezas de la infraestructura que había tenido NetBeans anteriormente. NetBeans IDE 5.0 introdujo un soporte mucho mejor para el desarrollo de nuevos módulos, el nuevo constructor intuitivo de interfaces Matisse, un nuevo y rediseñado soporte de CVS, soporte a Sun ApplicationServer 8.2, Weblogic9 y JBoss 4. Con Netbeans 6.1 y 6.5 se dio soporte a frameworks comerciales como son Struts, Hibernate.

1.3 DEFINICIÓN DE NETBEANS

NetBeans, el IDE (Entorno de Desarrollo Integrado) Java de Código Abierto.

1.3.1 ¿QUÉ ES NETBEANS?

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios (¡y subiendo!) en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio 2000 y continúa siendo el patrocinador principal de los proyectos. A día de hoy hay disponibles dos productos: el NetBeans IDE y el NetBeans Platform. El NetBeans IDE es un entorno de desarrollo - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java - pero puede servir para cualquier otro lenguaje de programación. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso. También disponible está el NetBeans Platform; una base modular y extensible usada como una estructura de integración para crear aplicaciones de escritorio grandes. Empresas independientes asociadas, especializadas en desarrollo de software, proporcionan extensiones adicionales que se integran fácilmente en la plataforma y que pueden también utilizarse para desarrollar sus propias herramientas y soluciones. Ambos productos son de código abierto y gratuito para el uso tanto comercial y como no comercial. El código fuente está disponible para su reutilización de acuerdo con la Common Development and Distribution License (CDDL).

1.3.2 PROCESO DE LIBERACIÓN DE VERSIONES DE NETBEANS

- Se debe lanzar una nueva versión aproximadamente cada seis meses, dos por año, esa es la Meta.
- Siempre se necesita estar mirando hacia la nueva versión y la siguiente. No todas las versiones son iguales. Se puede planear muchos cambios en una versión, pero mucho trabajo de estabilización en la siguiente. Eso no significa que se tolerará una versión con muchos errores, pero como NetBeans evoluciona, es predecible que ocasionalmente ciertas funcionalidades serán rescritas significativamente.
- A pesar que la agenda general dice que deben haber una nueva versión cada seis meses, las fechas exactas deben ser fijadas.
- La agenda de liberación de una nueva versión consiste en:
 1. Congelamiento de características. Esto se realiza después de un tiempo que no serán introducidas nuevas funcionalidades. Solo serán permitidos en el CVS cambios en el código fuente a causa de corrección de errores.
 - Se crea una rama en el CVS para agregar correcciones de errores y cambios en la documentación. La convención para el nombre de esta rama será en el CVS será releaseNN donde NN es la versión pendiente a liberar.
 - Después del congelamiento de características, los cambios en la interfaz de usuario deben ser mínimos. Cualquier cambio pendiente en la interfaz de usuario debe ser comunicado por adelantado y solo debe hacerse para corrección de errores.
 - Durante la etapa de estabilización, los binarios son marcados como NB X.Y beta. Construcciones diarias estarán disponibles para bajar y todos los usuarios están invitados a probar el software. Se espera que los desarrolladores respondan rápidamente a los reportes de errores. Los reportes de errores deben ser preferiblemente llenados en la base de datos de errores, pero los desarrolladores deben también monitorear los usuarios de las listas de distribución y responder las recomendaciones aquí escritas.

- Durante la etapa de estabilización, el README y las notas de la versión son escritas. Los primeros borradores de estos dos documentos críticos deben preferiblemente hacerlos antes que después.
2. Etapa de estabilización. En la cual los errores son localizados y arreglados.
 3. Release candidate 1. Si ningún problema serio es encontrado la última versión candidate a ser liberada se convierte en la versión final. Debe haber por lo menos una semana entre la última versión candidata y la versión final.
 - después que la primera versión candidate es hecha, todos los cambios en el código fuente deben ser negociados por adelantado escribiendo un post en la lista de distribución de `nbdev@netbeans.org` `nbdev 40 netbeans.org`.
 4. Release candidate 2 – La segunda versión candidate.
 5. Final release – La final, versión pública de la nueva versión de NetBeans

1.3.3 VERSIONES

Versiones NetBeans	
Versión	Fecha de Lanzamiento
NetBeans 6.5	19 de Noviembre del 2008
NetBeans 6.1	Abril 28 del 2008
NetBeans 6.0	Diciembre 3 del 2007
NetBeans 5.5.1	Mayo 24 del 2007
NetBeans 5.5	Octubre 30 del 2006
NetBeans 5.0	Enero 2006
NetBeans 4.1	Mayo 2005
NetBeans 4.0	Diciembre 2004
NetBeans 3.6	Abril 2004
NetBeans 3.5	Junio 2003

1.3.4 RADIOGRAFÍA

El NetBeans IDE 6.5 contiene aproximadamente 1.990.915 líneas de código de las cuales el 99.19% son de Java. El esfuerzo para producir un software de este tamaño según el modelo COCOMO es de aproximadamente 582 persona / año.

Lenguajes de programación utilizados en NetBeans IDE 6.5

Lenguaje	Líneas De Código	%
JAVA	1.974.732	99,19%
JSP	7.917	0.40%
Haskell	3.138	0,16%
CPP	1.761	0.09%
Yacc	1.123	0,06%
Sh	1.080	0,05%
Lex	506	0,03%
Perl	350	0,02%
Objc	288	0,01%
Ansic	20	0,00%

1.3.5 ESTADO ACTUAL DE NETBEANS IDE 6.5

Página Web	http://www.netbeans.org/
Inicio del Proyecto	1999
Versión Actual	6.5 19 de Noviembre de 2008
Líneas de Código Fuente	1.990.915
Esfuerzo estimado de desarrollo (persona-año / persona-mes)	582,15 / 6.985,81
Estimación de tiempo (años-meses)	6,02 / 72,23
Estimación del nº de desarrolladores en paralelo	96,71
Estimación de coste	\$ 78.640.629 (Salario promedio \$56,286/año con unos costos extras de 2,40)

II. CAPÍTULO

2.1 REQUERIMIENTOS E INSTALACIÓN DE NETBEANS IDE

2.1.1 REQUERIMIENTOS DEL SISTEMA

NetBeans IDE se ejecuta en sistemas operativos que apoyan el Java VM (Máquina Virtual) y se ha probado en las siguientes plataformas.

Los Sistemas operativos que soporta son:

- Windows XP SP2
- Sistema Operativo Linux Ubuntu 8.1
- Solaris OS versión 10

Antes de instalar el IDE, el Kit de Desarrollo Java SE (JDK) 5 Actualización 14 (versión 1.5.0_14) o posterior (incluyendo JDK 6 Actualización 4) debe estar instalado en su sistema. Si no tiene una instalación de JDK 5 Actualización 14 o posterior, usted no puede continuar con la instalación.

Este software puede encontrarse en la página Web de Sun <http://java.sun.com>.

2.1.2 OPCIONES DE DESCARGA DEL INSTALADOR

En la Página de Descarga de NetBeans IDE, usted puede descargar uno de varios instaladores, cada uno contiene el IDE base y herramientas adicionales.

Los siguientes instaladores que están disponibles:

Web & Java EE. Provee herramientas para desarrollar aplicaciones Java SE, Java EE y aplicaciones web. Esta opción de descarga también incluye el servidor GlassFish V2 UR2 y el software Apache Tomcat.

Mobility. Soporta el desarrollo de aplicaciones Java ME e incluye el paquete Mobility para ciertas aplicaciones MIDP para aparatos móviles. Notese que algunas características de Mobility, tales como el asistente para Conexión Inalámbrica (Wireless Connection), requiere características que están disponibles en el paquete Web & Java EE. Si usted planea usar estas características, descargue la opción de instalación "All".

Java SE. Provee todas las características estándares para el desarrollo Java SE. El soporte para Desarrollo de Extensiones NetBeans (NetBeans Plugin Development) también está incluido.

Ruby. Provee herramientas para el desarrollo Ruby y soporta Rails y JRuby.

PHP. Provee herramientas para el desarrollo PHP. La opción de descarga PHP está disponible en Acceso Reciente (Early Access).

C/C++. Soporta el desarrollo en lenguaje C/C++.

All. Esta es la opción de descarga completa para NetBeans IDE. Contiene todos los paquetes y rutinas disponibles para el IDE. Incluye herramientas para el desarrollo de Arquitectura de Servicio Orientado (Service Oriented Architecture) y construcción de diagramas de Modelado Unificado de Lenguaje (Unified Modeling Language).

Se puede descargar gratuitamente de la dirección **www.netbeans.org**.

La nota: La resolución de pantalla mínima del IDE es de 1024x768 pixeles.

2.1.3 CONFIGURACIONES RECOMENDADAS DE HARDWARE:

- **Microsoft Windows XP SP2 Profesional:**
 - **Procesador:** 2.6 GHz Intel Pentium IV o equivalente
 - **Memoria:** 1 GB
 - **Espacio en disco:** 1 GB espacio libre en disco

- **Sistema Operativo Linux Ubuntu 8.1:**
 - **Procesador:** 2.6 GHz Intel Pentium IV o equivalente
 - **Memoria:** 1 GB
 - **Espacio en disco:** 850 MB espacio libre en disco

- **Sistema Operativo Solaris:**
 - **Procesador:** 2.6 GHz Intel Pentium IV o equivalente
 - **Memoria:** 1 GB
 - **Espacio en disco:** 850 MB espacio libre en disco

2.1.4 CONFIGURACIONES MÍNIMAS DE HARDWARE:

- **Microsoft Windows XP SP2 Profesional:**
 - **Procesador:** 800MHz Intel Pentium III o equivalente
 - **Memoria:** 512 MB
 - **Espacio en disco:** 750 MB espacio libre en disco

- **Sistema Operativo Linux Ubuntu 8.1:**
 - **Procesador:** 800 MHz Intel Pentium III o equivalente
 - **Memoria:** 512 MB
 - **Espacio en disco:** 650 MB espacio libre en disco

- **Sistema Operativo Solaris:**
 - **Procesador:** AMD Opteron 1200 1.8 GHz
 - **Memoria:** 512 MB
 - **Espacio en disco:** 650 MB espacio libre en disco

2.2 INSTALACIÓN DE NETBEANS IDE

A continuación se darán las instrucciones de como instalar NetBeans IDE (Standalone Version) en cada una de las plataformas soportadas. Recuerde que el JDK de java ya debe estar instalado en el sistema.

El lanzamiento de NetBeans IDE 6.5 provee varias características y mejoras nuevas, tales como mejores características de edición JavaScript, soporte para usar estructuras Spring de soporte web, integración MySQL más ajustada y una mejor manera de compartir librerías entre proyectos dependientes. El aclamado soporte para Ruby/JRuby ha sido mejorado con un nuevo editor de soluciones rápidas (Quick Fix), un administrador para la plataforma Ruby, soporte para depuración rápida en JRuby y muchas otras nuevas características y soluciones.

Esta versión también provee mejoras en el rendimiento, especialmente en el inicio más rápido (hasta 40% más rápido), menor consumo de memoria y mejores respuestas mientras se trabaja con proyectos muy grandes.

2.2.1 INSTALACIÓN EN WINDOWS

Después de haber bajado el instalador, dar doble clic sobre el icono para iniciar el instalador.

1. Después de abrir el .exe toma un tiempo en configurar el instalador y luego nos da la opción de personalizar la instalación pudiendo agregar o eliminar paquetes.

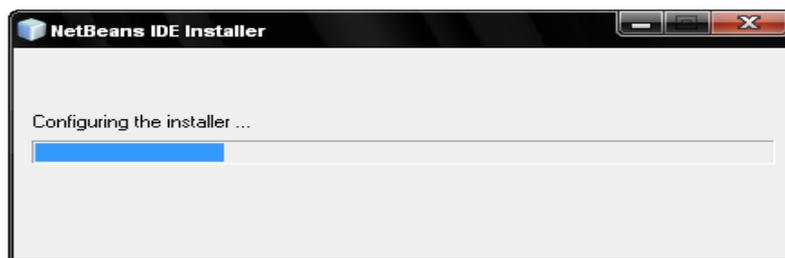


Figura. 1. Configurando el instalador.



Figura. 2. Bienvenido al Instalador de NetBeans en Windows

2. Después aparecerá el Acuerdo de Licenciamiento (License Agreement), acéptelo y de clic en Siguiente (Next).

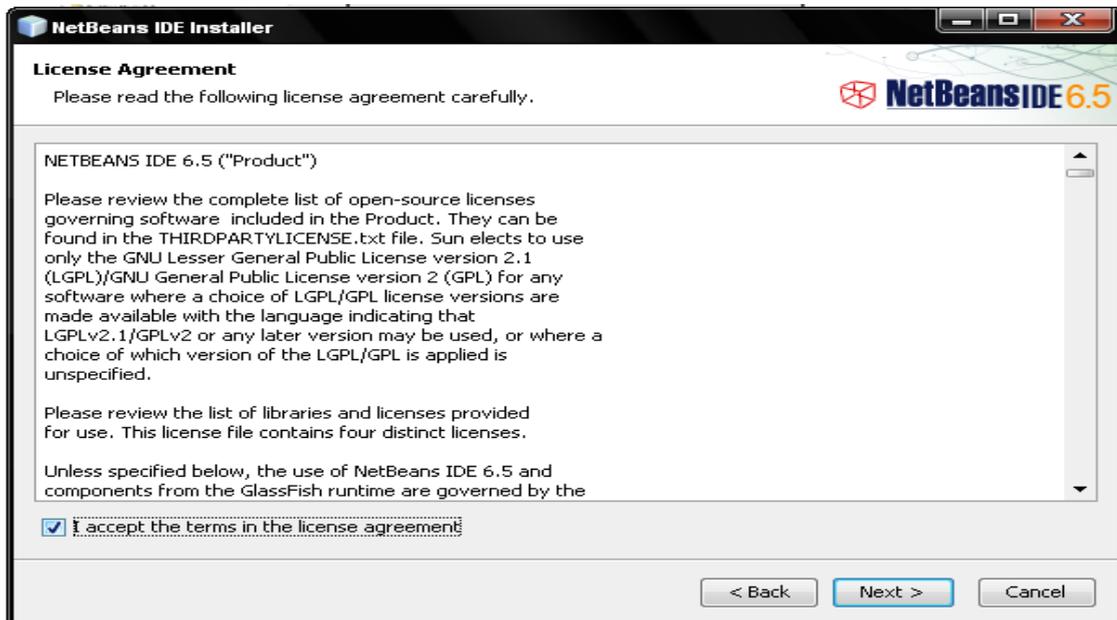


Figura. 3. Acuerdo de Licenciamiento en Windows

3. Especificar un directorio vacío donde instalar NetBeans IDE. Dar clic en Siguiente (Next).

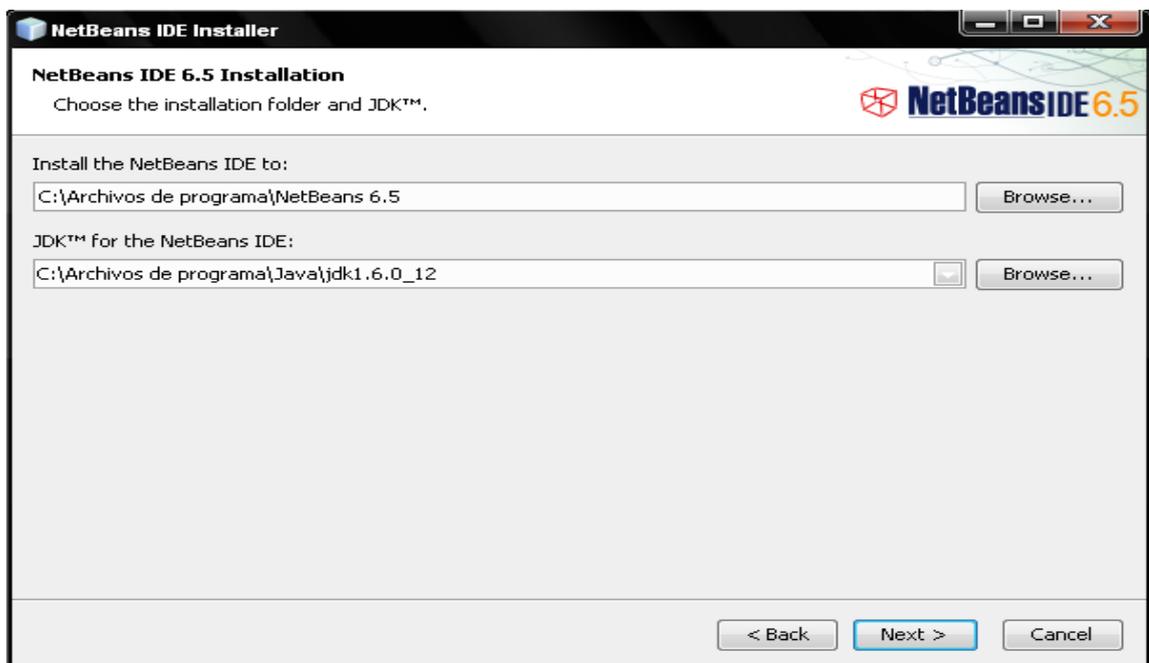


Figura 1. Especificar directorio donde instalar NetBeans IDE en Windows

4. Seleccionar el JDK que se quiere usar junto con el NetBeans IDE de la lista de selección si hay más de uno. Dar clic en Siguiente (Next).

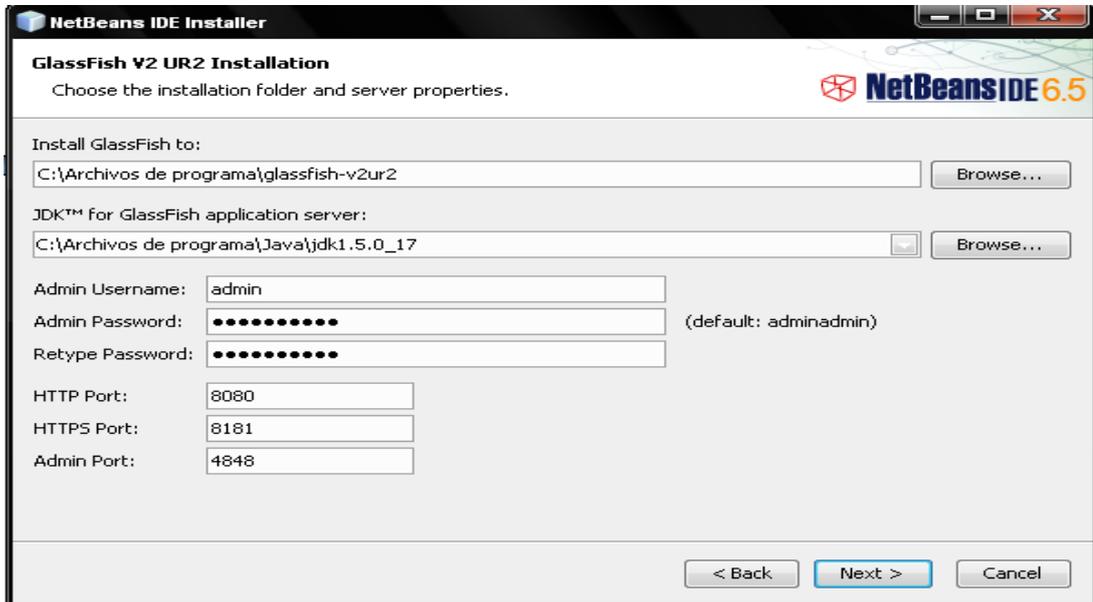


Figura. 4. Seleccionar el JDK junto con el NetBeans IDE

5. Verifique que el directorio de instalación es el correcto y que se cuenta con espacio suficiente en el sistema para la instalación.

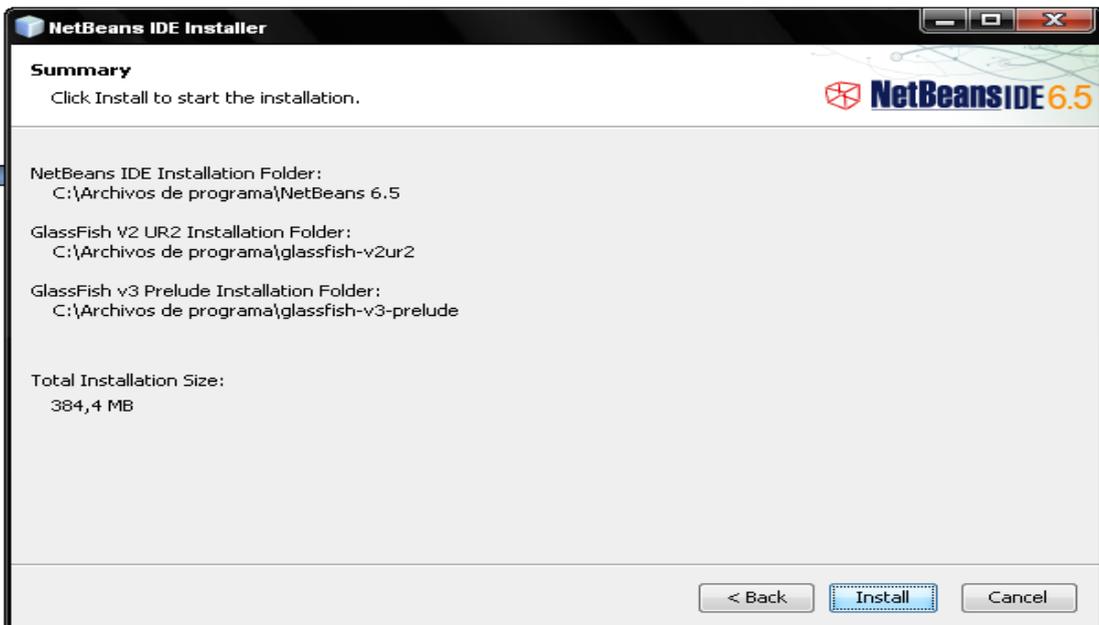


Figura. 5. Resumen para instalación de NetBeans IDE en Windows

6. De clic en Siguiente (Next) para iniciar la instalación.

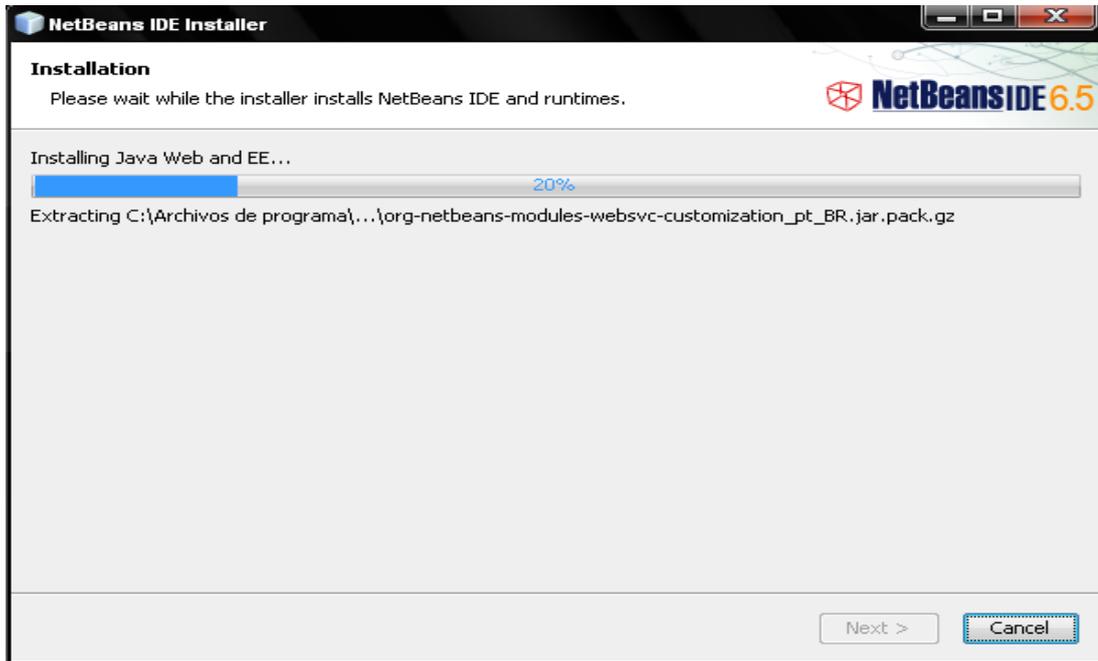


Figura. 6. Instalación de NetBeans IDE en Windows

7. Esperar mientras finaliza la instalación de NetBeans.

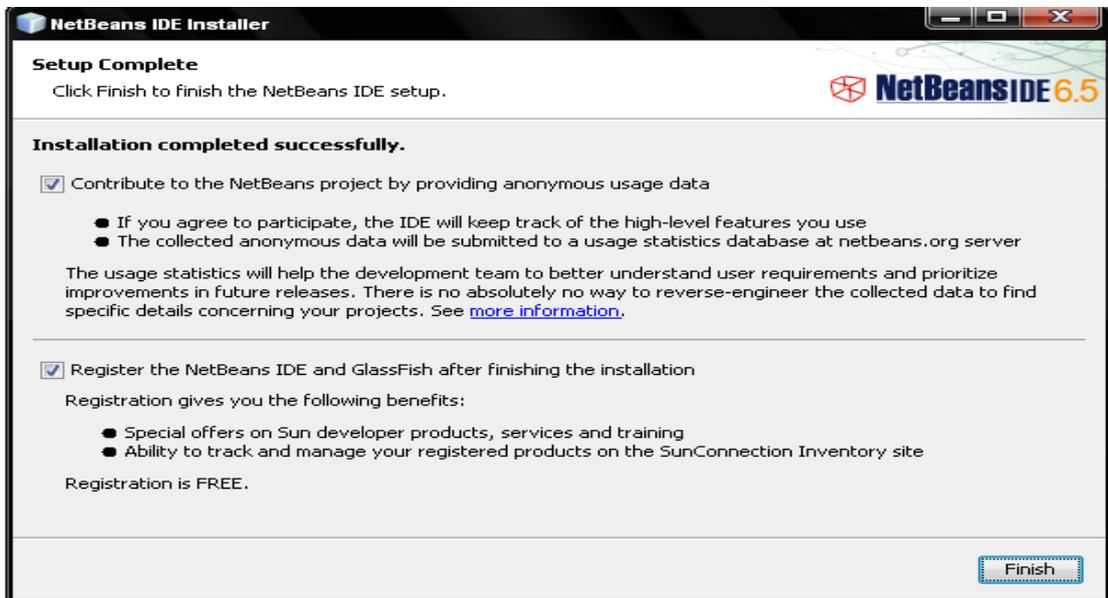


Figura. 7. Instalación completa de NetBeans IDE en Windows

2.2.2 INSTALACIÓN EN LINUX UBUNTU 8.1

Para descargar el instalador de NetBeans 6.5 para Ubuntu, la encontramos en la siguiente dirección Web:

<http://services.netbeans.org/bouncer/index.php?product=netbeans-6.5-java&os=linux>

1. Navegue hacia el directorio que contiene el instalador.
2. Si es necesario, cambie los permisos del instalador para hacerlo ejecutable escribiendo lo siguiente en la línea de comando

```
chmod +x netbeans-6.5-ml-java-linux.sh
```

3. Ejecute el instalador escribiendo lo siguiente en la línea de comando

```
./netbeans-6.5-ml-java-linux.sh
```

4. Una vez que se ejecuta el comando seguimos con la instalación de NetBeans y pulsamos siguiente (Next):

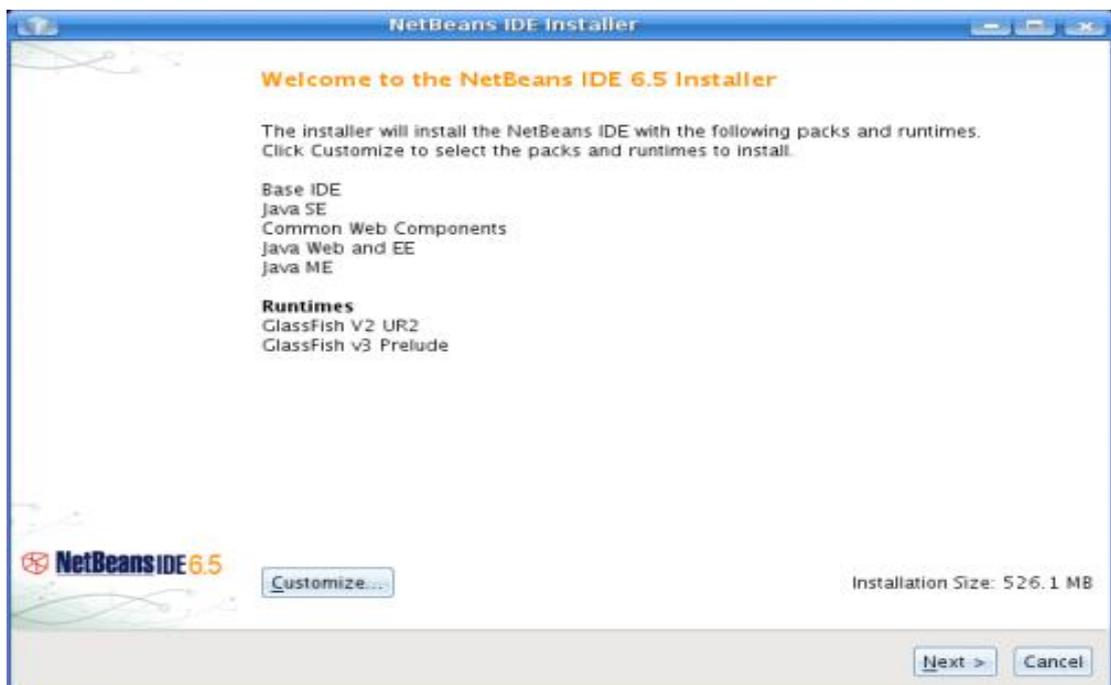


Figura. 8. Bienvenido al instalador de NetBeans IDE en Linux (Ubuntu)

5. Después aparecerá el Acuerdo de Licenciamiento (License Agreement), acéptelo y de clic en Siguiente (Next).

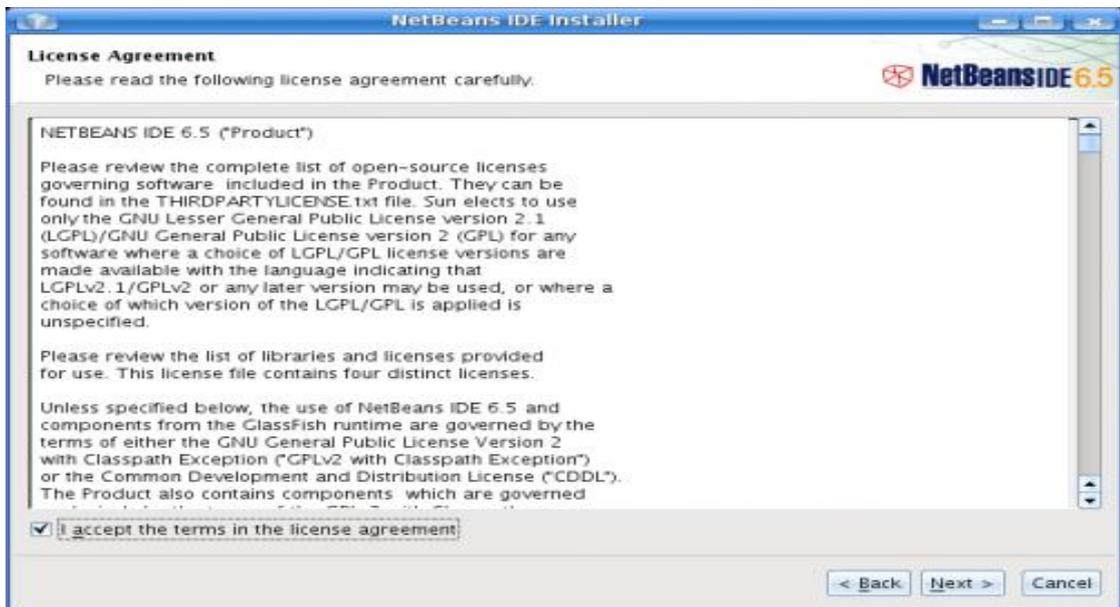


Figura. 9. Acuerdo de Licenciamiento para Linux (Ubuntu)

6. Especificar un directorio vacío donde instalar NetBeans IDE y que versión de JDK se va a usar. He instalado NetBeans en mi carpeta home y no a carpeta por defecto de Ubuntu.

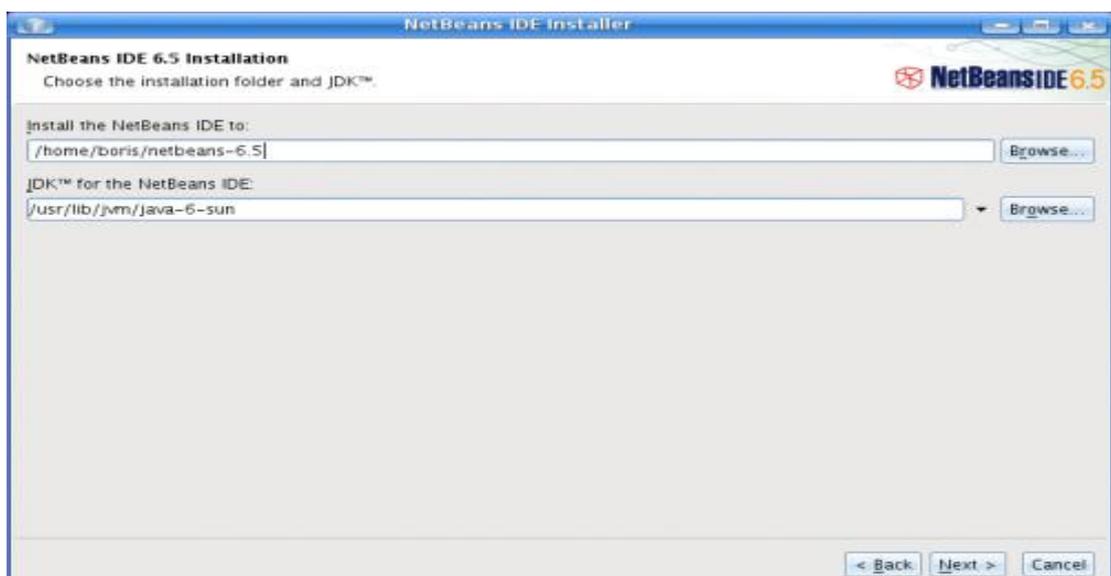


Figura. 10. Especificar donde instalar NetBeans IDE en Linux (Ubuntu)

7. Especificar carpeta para servidores de aplicaciones Glassfish v2 y que versión de JDK usara. Dar clic en siguiente (Next).

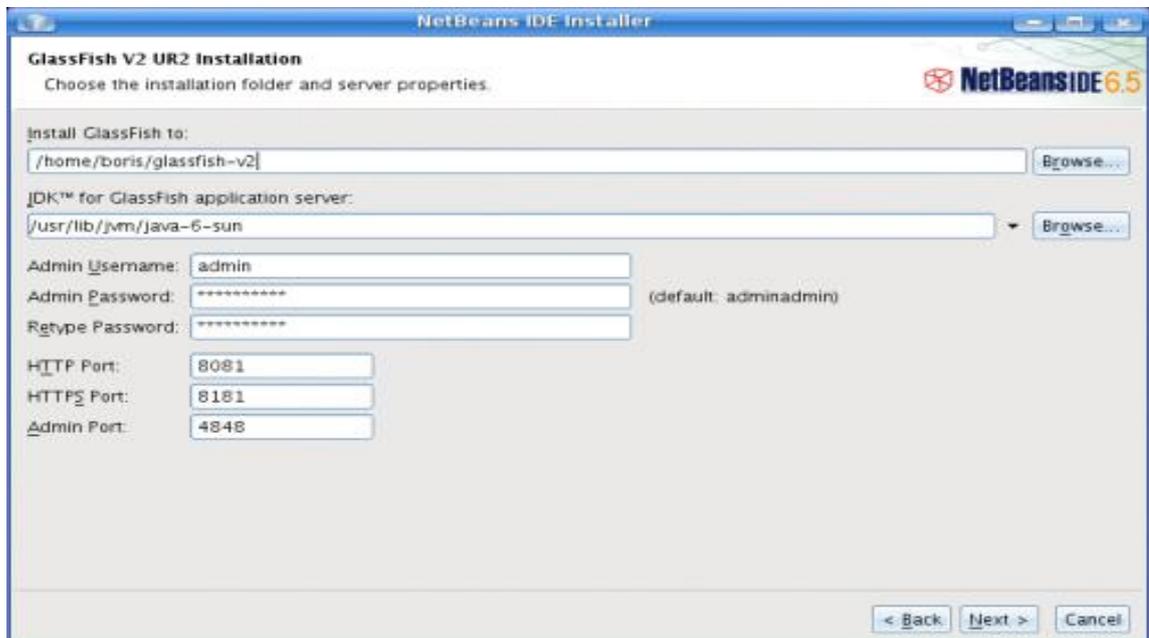


Figura. 11. Especificar servidores y versión de JDK para Linux (Ubuntu)

8. Y especificar carpeta para Glassfish v3 y pulsamos siguiente (Next).

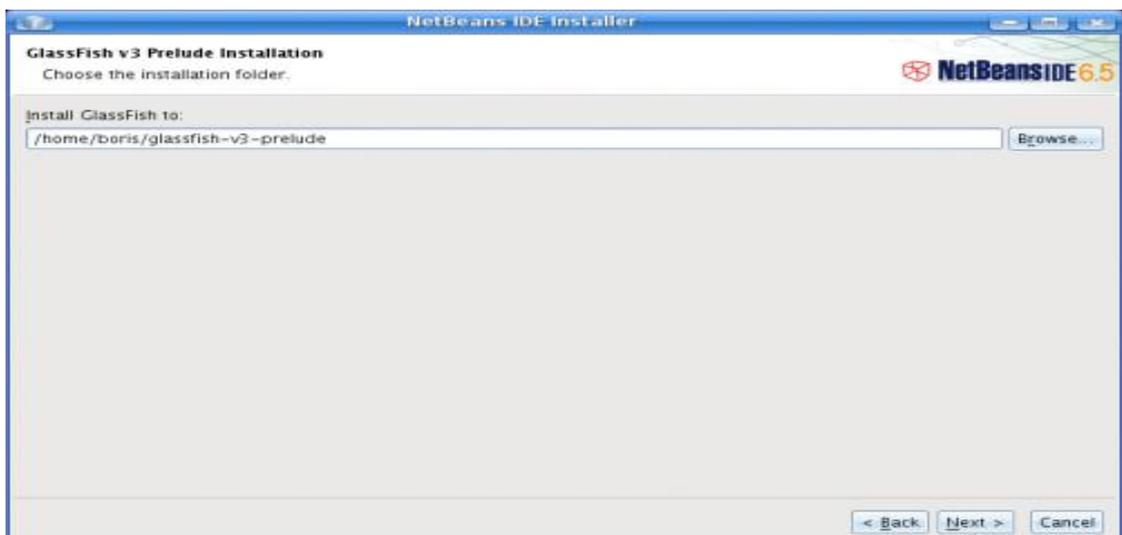


Figura. 12. Especificar carpeta para Glassfish

9. Antes de seguir, verifique que el directorio de instalación es el correcto y que se cuenta con espacio suficiente en el sistema para la instalación. Pulse Install para continuar con la instalación:



Figura. 13. Resumen para instalación de NetBeans en Linux (Ubuntu)

10. Y ahora nos queda esperar unos minutos para que el instalador copie todos los archivos necesarios:

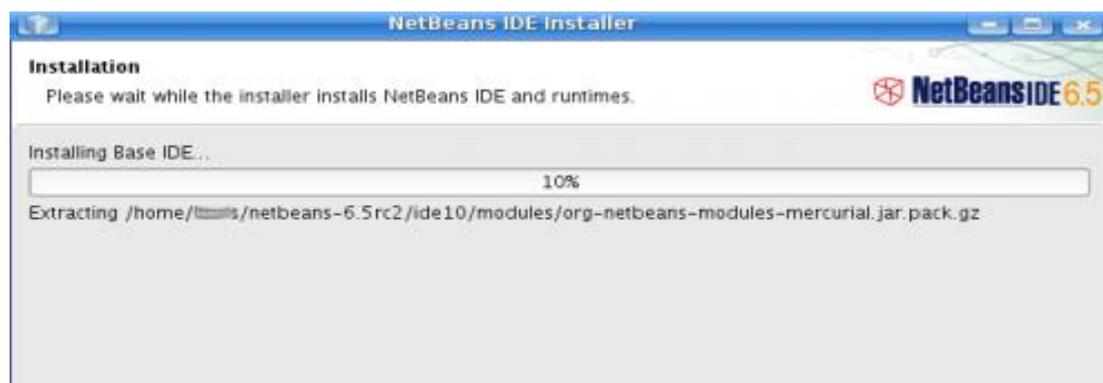


Figura. 14. Instalación de NetBeans IDE en Linux (Ubuntu)

11. Cuando la instalación este completa y podemos registrarnos.

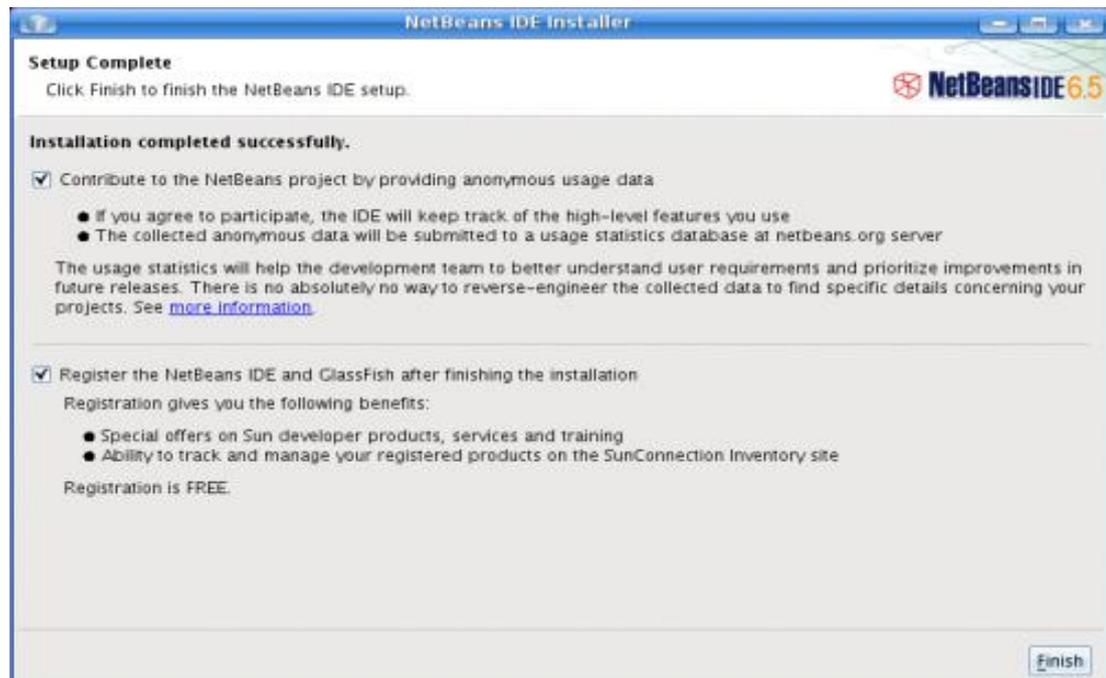


Figura. 15. Instalación completa de NetBeans IDE en Linux (Ubuntu)

12. Ya finalizada la instalación, están listos para empezar a usar NetBeans 6.5.



Figura. 16. Pantalla principal de NetBeans IDE 6.5

2.2.3 INSTALACIÓN EN SISTEMAS OPERATIVOS SOLARIS

1. Navegue hacia el directorio que contiene el instalador.
2. Si es necesario, cambie los permisos del instalador para hacerlo ejecutable escribiendo lo siguiente en la línea de comando (`netbeans_installer` es el nombre del instalador de NetBeans que se bajó): `$ chmod +x netbeans_installer`.
3. Ejecute el instalador escribiendo lo siguiente en la línea de comando (De nuevo, `netbeans_installer` es el nombre del instalador de NetBeans que se bajó): `$./netbeans_installer`.
4. Después aparecerá el Acuerdo de Licenciamiento (License Agreement), acéptelo y de clic en Siguiente (Next).
5. Especificar un directorio vacío donde instalar Netbeans IDE. Dar clic en Siguiente (Next).
6. Selecciones el JDK que se quiere usar junto con el NetBeans IDE de la lista de selección si hay más de uno. Dar clic en Siguiente (Next).
7. Verifique que el directorio de instalación es el correcto y que se cuenta con espacio suficiente en el sistema para la instalación.
8. De clic en Siguiente (Next) para iniciar la instalación.
9. Y luego finalizar el instalador

III. CAPÍTULO

3.1 ESTUDIO Y REVISIÓN DE NETBEANS

3.1.1. INTRODUCCIÓN

Este manual proporciona una simple y rápida introducción al flujo de trabajo de NetBeans IDE. Culminado este manual, tendrá un conocimiento general de cómo crear, construir y ejecutar aplicaciones en el IDE.

NetBeans es un entorno de desarrollo integrado (IDE) que permite editar programas en java, compilarlos, ejecutarlos, depurarlos, construir rápidamente un interfaz gráfico de una aplicación eligiendo los componentes de una paleta, etc.

3.1.1.1 Conceptos Básicos del IDE

Antes de empezar a adentrarnos en este proyecto tomemos un minuto para conocer algunos de los conceptos básicos que se involucran en el uso del IDE.

3.1.1.2 La Ventana Files

El punto de partida para el desarrollo en el IDE es la ventana Files. La ventana Files es donde usted organiza el contenido de su proyecto, accede y ejecuta comandos en archivos individuales, y ve la estructura de la fuente de sus archivos. La ventana Files contiene todos los directorios, paquetes, y archiva archivos que usted ha agregado a su proyecto.

Cuando usted primero ejecuta el IDE, la ventana Files contiene el directorio de ejemplo de NetBeans con algún código de ejemplo. Cada fuente de archivo tiene su propio nodo en la ventana Files. Estos nodos tienen:

- **Comandos del Menú Contextual** - Puede ejecutar comandos en los archivos pulsando click derecho y escogiendo del menú contextual. Los comandos disponibles, varían dependiendo del tipo de nodo con los que está trabajando.
- **Propiedades** – *Choose Window > Properties* (Ctrl-1) para abrir un ventana de Propiedades. La ventana de Propiedades siempre muestra las propiedades del componente del enfoque de la interfaz de usuario en NetBeans.
- **Subnodos** - Usted puede extender muchos nodos en la ventana Filesystems para ver subnodos que representa la estructura interna del archivo del nodo. Estos subnodos tienen a menudo sus propias propiedades y comandos en el menú contextual.

La ventana de Files no muestra un nodo para cada archivo en un directorio montado. Por ejemplo, para compilar el form object ColorPicker, la fuente del directorio contiene el ColorPicker.java fuente del archivo, el archivo ColorPicker.form usado para construir su GUI en el IDE, y compilar la clase ColorPicker.class. El IDE esconde los archivos .form y .class por defecto, tan sólo un nodo se muestra para ColorPicker. Usted también puede escoger esconder los archivos por el tipo y extensión.

3.1.1.3 Los proyectos en el IDE

Un proyecto es la unidad básica de trabajo en el IDE. Incluye todos los archivos con qué está trabajando y las configuraciones de IDE que aplica a esos archivos. NetBeans IDE tiene un sistema de los proyectos muy claro. Usted siempre tiene abierto un proyecto a la vez. Todo lo que está en la ventana Files es parte del proyecto abierto actualmente.

NetBeans siempre trabaja sobre proyectos, no puede compilar ficheros que no estén integrados dentro de un proyecto.

Los proyectos NetBeans se basan en Ant pero no es necesario conocer Ant para manejarlos.

Una estructura típica de un directorio de un proyecto NetBeans incluye los siguientes subdirectorios:

- **build:** donde se sitúan los ficheros .class compilados
- **dist:** donde se sitúan el fichero empaquetado .jar
- **nbproject:** incluye la información del proyecto NetBeans y generalmente no debe tocarse
- **src:** donde se incluyen los fuentes
- **test:** donde se incluyen los fuentes de los tests JUnit para realizar pruebas de unidad

La herramienta provee de asistentes para empezar proyectos desde cero o para crear un proyecto con fuentes ya existentes

Nota: No tiene que crear un nuevo proyecto para cada aplicación en la que va a trabajar. Puede agregar la fuente de los archivos para múltiples aplicaciones en la ventana Files y trabajar enseguida en ellos. Cada proyecto, sin embargo, tiene sólo un classpath y control de configuraciones IDE.

La mayoría de las configuraciones de IDE es aplicado a uno de dos niveles: para el proyecto entero o para archivos individuales. Las configuraciones de proyectos amplios son manejadas en la ventana de Opciones. Puede abrir la ventana de Opciones escogiendo *Tools > Options* en la ventana Principal. Establecerá la configuración en un archivo usando la ventana de Propiedades.

3.1.1.4 Ventana Servicios

La ventana de Servicios es el punto de entrada principal a los recursos del runtime. Muestra una vista lógica de recursos importantes del runtime como los servidores, bases de datos, y servicios web que son registrados con el IDE. Podemos abrir la ventana de Servicios escogiendo la Windows> Services (Ctrl-5).

3.1.1.5 Los Recursos de Runtime

- **Servidores**

Detalla todos los servidores actualmente registrado con el IDE. Cualquier servidor se instala automáticamente en el IDE en el proceso de instalación del mismo. Use el IDE para registrar todos los otros servidores.

- **Bases de datos**

Detalla los drivers de la base de datos que se han agregado al IDE. Otros nodos pueden aparecer, dependiendo de los módulos que se instalan en el IDE.

3.2 CONOCIMIENTO BÁSICO DE LAS HERRAMIENTAS

3.2.1. TIPOS DE DATOS

Tipo	Características	Utilización
byte	8 bits-complemento a 2	
short	16 bits-complemento a 2	
int	32 bits-complemento a 2	
long	64 bits-complemento a 2	
float	32 bits-IEEE 754	
double	64 bits-IEEE 754	
boolean	Valores true y false	
char	16 bits-carácter	
String	Se trata como una clase No se pueden cambiar los caracteres que lo forman	String s; S="Cadena" S=new String("Cadena");

3.2.2. OPERADORES

Tipo	Operador	Descripción
Aritméticos	+, -, *, / y %	Suma, Resta, Producto, División y Resto
	++, --	Incremento, Decremento
Relacionales	<, <=	Menor que, Menor o igual que
	>, >=	Mayor que, Mayor o igual que
	==, !=	Distinto de
Condicionales	&	AND
		OR
	&&	AND Condicional
		OR Condicional
	!	NOT
Manejo de bits	>>, <<	Desplazar a la derecha, Desplazar a la izquierda
	>>>>	Desplazamiento sin signo
	&, , ^	AND, OR, XOR

3.2.3. ESTRUCTURAS DE CONTROL

En esta sección conocerá como construir estructuras selectivas, ciclos repetitivos y expresiones en JAVA, que pueda compilar y ejecutar programas con NetBeans IDE y así aprender a adicionar nuevos programas a un proyecto.

3.2.3.1. Sentencia if

Permite a un programa tomar una decisión para ejecutar una acción u otra, basándose en el resultado verdadero o falso de una expresión.

```
import javax.swing.JOptionPane;

    public class CMenor {

        //menor de 3 numeros a, b y c

        public void main(String []args){

            float a, b, c, menor;

            String primerNumero, segundoNumero, tercerNumero;

            //Leer los valores de a, b, y c

            primerNumero = JOptionPane.showInputDialog("Digite el primer número: ");

            segundoNumero = JOptionPane.showInputDialog("Digite el segundo número: ");

            tercerNumero = JOptionPane.showInputDialog("Digite el tercer número: ");

            a = Float.parseFloat(primerNumero);

            b = Float.parseFloat(segundoNumero);

            c = Float.parseFloat(tercerNumero);
```

```

    if (a < b){
        if (a < c)
            menor = a;
        else
            menor = c;
    }
    else {
        if (b > c)
            menor = b;
        else
            menor = c;
    }
    System.out.println("Menor:"+menor);
} //fin de main
} // fin de clase

```

3.2.3.2. Sentencia WHILE

La sentencia WHILE ejecuta una sentencia simple o compuesta, cero o más veces, dependiendo del valor de una expresión lógica.

```

public class While1{
    public static void main(String[] args){

```

*/*Vamos a hacer un programa que nos muestre los números pares del 1 al 100, y vamos hacer uso de mod o modificador, que se usa con el signo de " % (porcentaje)" , el uso de él es así 4%2==0 aquí hago una división de 4 entre 2 pero no me interesa el resultado, sino el residuo de este. */*

```
int numero=0;
```

```
while(numero<=100){
```

```
//dentro del while evaluamos la siguiente condición
```

```
//si se cumple entrara y ejecutara lo siguiente.
```

```
    if(numero%2==0){
```

```
        System.out.println(numero);
```

```
//Impresión del número
```

```
    }
```

```
        numero++;
```

```
//y lo debemos incrementar para que no se cicle este while y
```

```
//haga lo que queremos que haga
```

```
    }
```

```
}
```

```
}
```

3.2.3.3. Sentencia do/while

Esta sentencia no es lo mismo que el while ya que este se ejecuta de una manera diferente, por lo menos se ejecutara una vez, así es, el do/while tiene esa condición que pase lo que pase una vez será ejecutado, y si se cumple la condición se volverá a ejecutar.

```
public class DoWhile1{  
  
    public static void main(String[] args){  
  
        int numero=0;  
  
        /*Vamos a comprobar el uso de Do/While, se darán cuenta que la variable numero  
        es iniciada en 0 (cero) así que no cumple la condición (la que está dentro del  
        while) de ser mayor que 0 (cero), pero como hemos dicho el do/while primero  
        ejecuta luego evalúa, así que primero entrara y incrementara la variable numero, y  
        ahora si entrara al while y así hasta llegar a 9. */  
  
        do {  
  
            System.out.println(numero);  
  
            numero++;  
  
        }while(numero>0 && numero<10);  
  
    }  
  
}
```

3.2.3.4. Sentencia For

Es una sentencia de control parecida a los while y do/while, pero tiene una diferencia, esta inicializa su variable de control, la evalúa y la repite de forma creciente o decreciente.

```
public class For1{  
    public static void main(String[] args){  
        //mostraremos una columna de números del 1 al 10  
        for(int i=0;i<10;i++){  
            System.out.println(i);  
        }  
        System.out.println();  
        //mostramos una columna de números del 10 al 1  
        for(int i=10;i>0;i--){  
            System.out.println(i);  
        }  
        for(int i=0;i<=10;i++){  
            System.out.println();  
            for(int j=0;j<i;j++){  
                System.out.print(i);  
            }  
        }  
        System.out.println();  
    }  
}
```

3.2.3.5. Sentencia SWITCH

La función switch permite ejecutar una de varias acciones, en función de una expresión. Es una sentencia especial para decisiones múltiples.

```
public class ciclos{

    public static void main (String args[]){

        int a=1;

        for(int num_dia=1; String dia; num_dia<12;num_dia++){

            switch(num_dia){

                case 1 : dia="Lunes"; break;

                case 2 : dia="Martes"; break;

                case 3 : dia="Miércoles"; break;

                case 4 : dia="Jueves"; break;

                case 5 : dia="Viernes"; break;

                case 6 : dia="Sábado"; break;

                case 7 : dia="Domingo"; break;

                default : dia="¿Que Día es Hoy?"; break;

            }// fin del switch

            System.out.println ("Hoy es"+dia+"el dia N°"+num_dia+"de la Semana");

        }//fin del for

    }// fin del main

}//fin de la clase
```

3.3 VENTAJAS Y DESVENTAJAS DE NETBEANS

3.3.1 VENTAJAS

- Facilidad de uso con Windows.
- Perfecto entorno de desarrollo.
- Facilidad de uso.
- Software libre.
- Es muy fácil crear un proyecto, no como con otros programas que te vuelven loco.
- Puedes ver todas las variables del programa en tiempo de ejecución.
- Los errores son bastante descriptivos, ayuda mucho las indicaciones que te da.

3.3.2 DESVENTAJAS

- Consume bastantes recursos aunque no demasiados.
- No posee interfaz para crear botones y similares.
- Come casi todos los recursos del ordenador.
- La barra de menú es muy extensa.

3.4 PLANIFICACIÓN Y ELABORACIÓN DEL MANUAL DE NETBEANS

Este manual tiene un objetivo primordial dar a conocer esta herramienta, la cual nos ayuda a trabajar en java, inicialmente promovido para que el aprendiz conozca un poco más sobre el IDE NetBeans 6.5.

3.4.1 MENÚ FILE

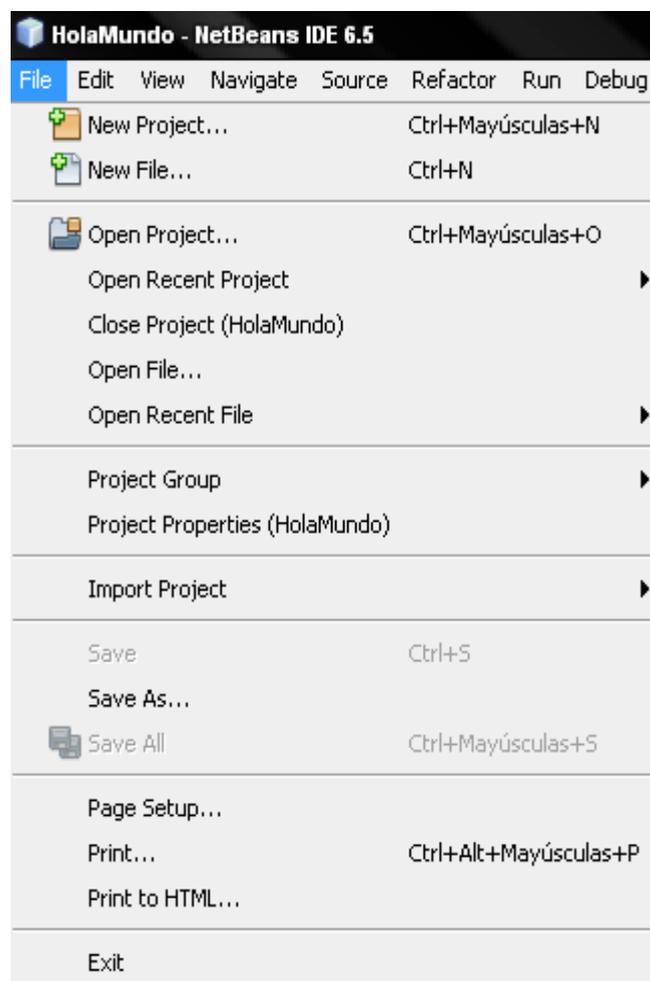


Figura 3. 1. Menú FILE

3.4.1.1 Creación de un Proyecto

Para crear un programa de consola en Java utilizando **NetBeans 6.5** lo primero que hay que hacer es crear un proyecto. Un proyecto nos permite administrar los archivos con el código fuente y compilado de una aplicación. Para crear un proyecto se sigue el siguiente procedimiento:

1. Ejecute el programa **NetBeans 6.5**. Al hacerlo aparecerá la ventana principal del programa como se ilustra en la figura 3.1.

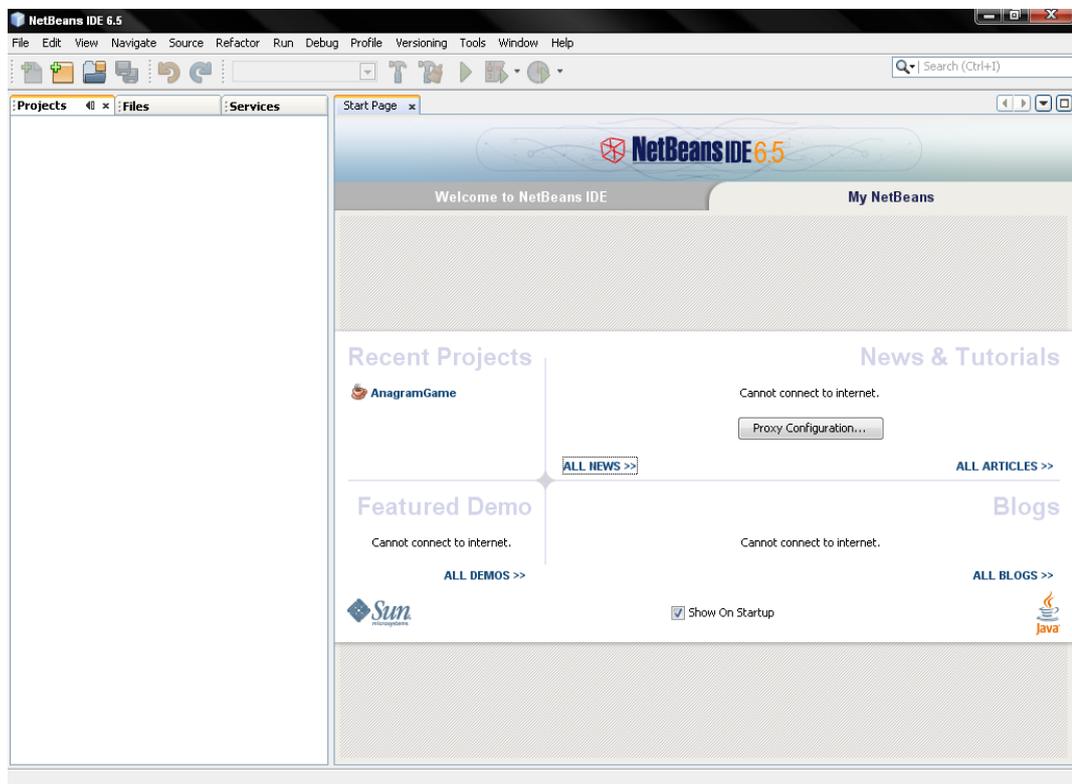


Figura 3. 2. Pantalla principal de NetBeans 6.5

2. Del menú principal de NetBeans 6.5, figura 3.2, seleccione la opción **File/New Project ...** , presione las teclas **Ctrl+Mayúsculas+N** o haga clic en el icono **New Project** mostrado en la figura 3.3.

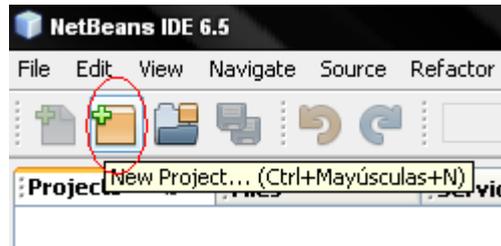


Figura 3. 3. Icono de New Project

3. Aparecerá la primera ventana del asistente para crear un nuevo proyecto, figura 3.3.

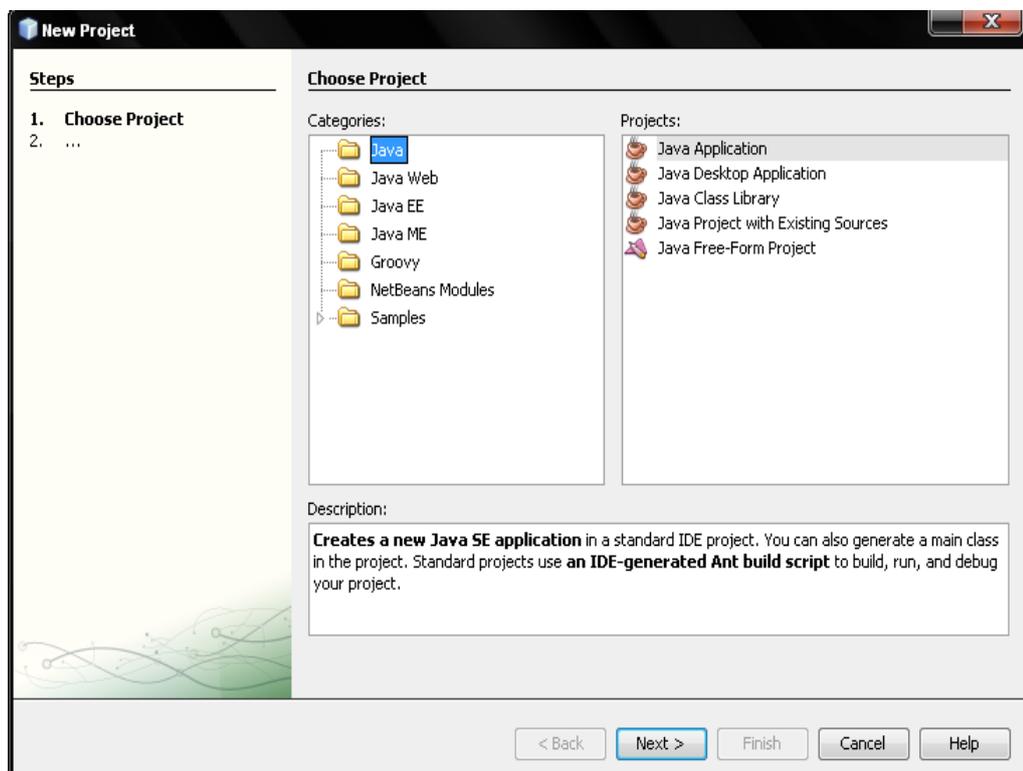


Figura 3. 4. Ventana New Project

En esta ventana del asistente seleccionaremos el tipo de proyecto que deseamos crear. Como vamos a crear una aplicación de consola, seleccionaremos la opción **General** en el recuadro **Categories:** y la opción **Java Application** en el recuadro **Projets** y luego presionaremos el botón **Next>**.

Las categorías que encontramos en la creación de un nuevo proyecto, son los siguientes:

3.4.1.1.1 Java (Figura 3.4)

- **Java Application:** Crea una nueva aplicación Java SE en un proyecto de IDE estándar. También puede generar una clase principal en el proyecto. Los proyectos estándar usan un **IDE-generated Ant build script** para construir, ejecutar y depurar su proyecto.
- **Java Desktop Application:** Crea un esqueleto de una aplicación del escritorio basado en el Armazón de Aplicación de Balance (JSR 296). Esta plantilla proporciona la infraestructura de la aplicación básica como una barra del menú, persistiendo la ventana de estado y barra de estado. Con esta plantilla, puede generar también el código para crear una interfaz de GUI para una mesa de la base de datos.
- **Java Class Library:** Crea una nueva librería Java SE en un proyecto de IDE estándar. Una librería Java SE no contiene una clase principal. Los proyectos estándar usan un **IDE-generated Ant build script** para construir, ejecutar y depurar su proyecto.
- **Java Project with Existing Sources:** Importa una aplicación de Java existente en un proyecto de IDE estándar. La aplicación de Java puede tener múltiples carpetas de la fuente. Los proyectos estándar usan un **IDE-generated Ant build script** para construir, ejecutar y depurar su proyecto.
- **Java Free-Form Project:** Importa una aplicación de Java existente y su escritura script en un proyecto de IDE. En los proyectos del libre-formulario,

IDE usa los comandos Ant script que proporciona para correr, compile, y depurar su proyecto. Cualquier modificación de la construcción del proceso puede manejarse directamente en su Ant script. Para hacer mantenimiento a largo plazo del proyecto más fácil, considere usar el Proyecto de Java con Fuentes Existentes en proyectos de plantilla.

3.4.1.1.2 Java Web

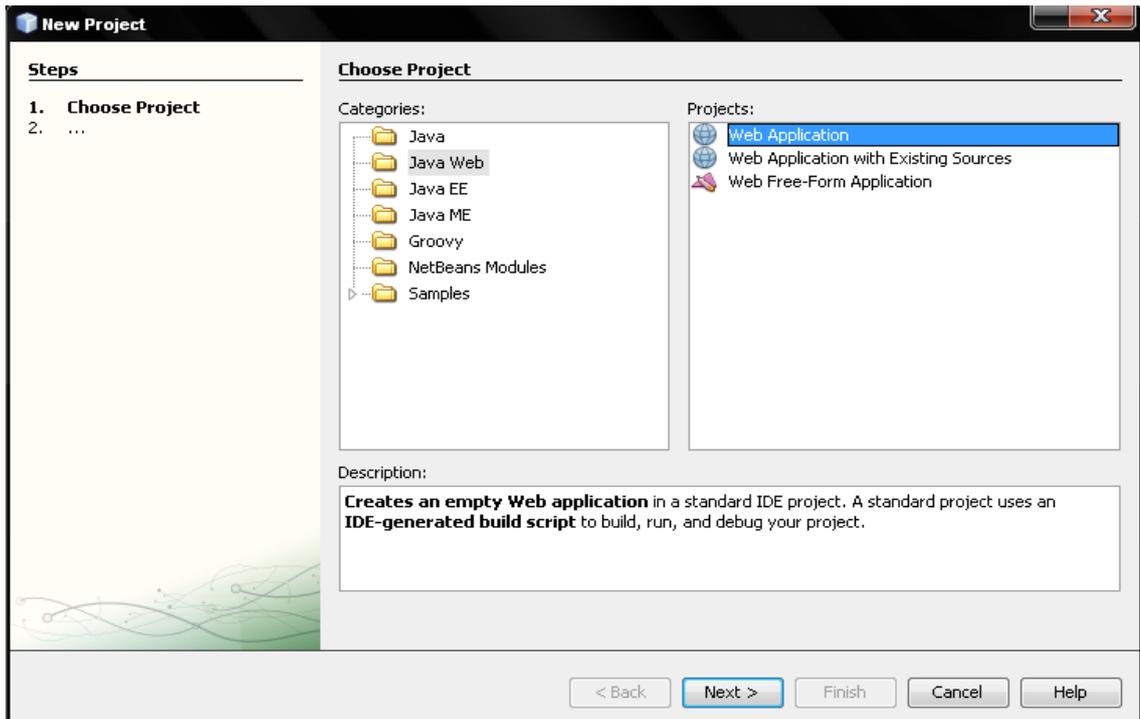


Figura 3. 5. Ventana New Project Java Web

- **Web Application:** Crea una aplicación Web vacía en un proyecto de IDE estándar. Los proyectos estándar usan un **IDE-generated Ant build script** para construir, ejecutar y depurar su proyecto.
- **Web Application with Existing Sources:** Importa una aplicación de Web existente en un proyecto web estándar. Los proyectos estándar usan un **IDE-generated Ant build script** para construir, ejecutar y depurar su proyecto.

- **Web Free-Form Application:** Importa una aplicación de Web existente y su script en un proyecto IDE. En los proyectos del libre-formulario, IDE usa comandos de Ant script que proporciona para correr, compilar, y depurar su proyecto. Cualquier modificación de la construcción del proceso puede manejarse directamente en su Ant script. Para hacer mantenimiento a largo plazo del proyecto más fácil, considere usar la Aplicación Web con Fuentes Existentes en proyectos de plantilla.

3.4.1.1.3 Java EE

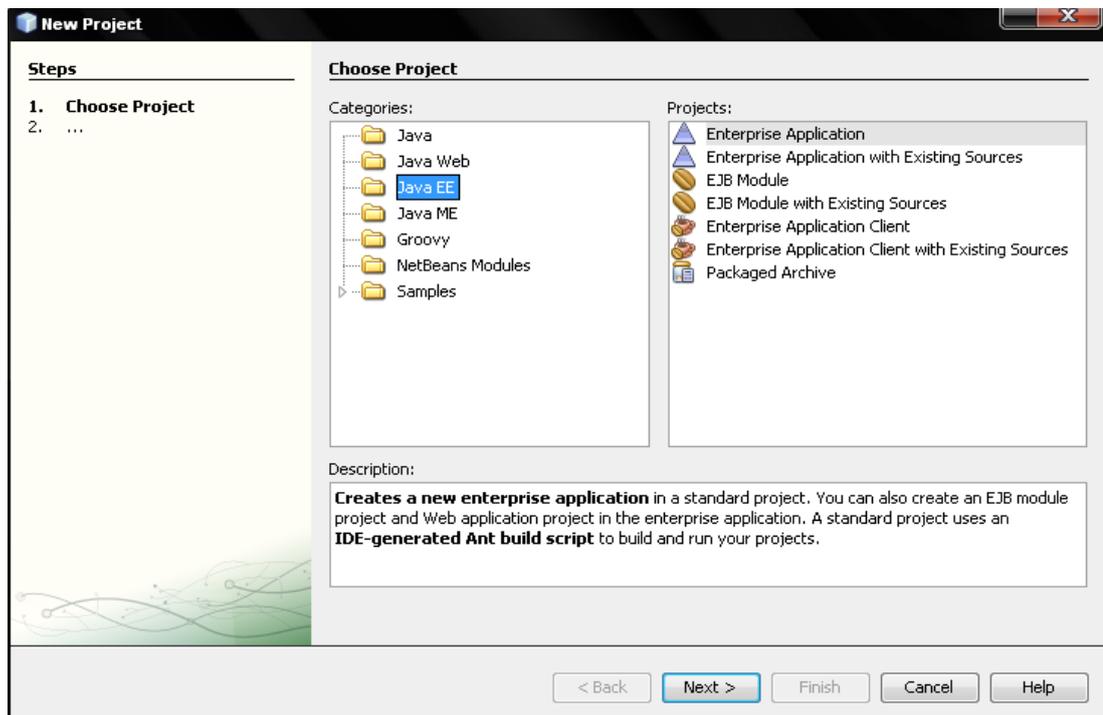


Figura 3. 6. Ventana New Project Java EE

- **Enterprise Application:** Crea una nueva aplicación Enterprise en un proyecto estándar. También puede crear un proyecto modulo EJB y un proyecto de aplicación Web en la aplicación Enterprise. Los proyectos estándar usan un **IDE-generated Ant build script** para construir y ejecutar su proyecto.
- **Enterprise Application with Existing Sources:** Las Importa una aplicación Enterprise existente en un proyecto de IDE estándar. La aplicación Enterprise debe conformar para los J2EE Cianotipos Proyecto Convenciones para las Aplicaciones Enterprise (vea

<http://java.sun.com/blueprints/code/projectconventions.html>). Este asistente también importa componentes que se anidan en la aplicación enterprise. Los proyectos estándar usan un **IDE-generated Ant build script** para construir y ejecutar su proyecto.

- **EJB Module:** Crea una nueva JavaBean Enterprise (EJB) en un proyecto de IDE estándar. Los proyectos estándar usan un **IDE-generated Ant build script** para construir y ejecutar su proyecto.
- **EJB Module with Existing Sources:** Importa un módulo existente Enterprise JavaBean (EJB) en un proyecto de IDE estándar. El módulo EJB puede tener cualquier estructura con tal de que tenga un archivo existente `ejb-jar.xml`. Los proyectos estándar usan un **IDE-generated Ant build script** para construir y ejecutar su proyecto.
- **Enterprise Application Client:** Crea una nueva aplicación enterprise proyecto cliente en un proyecto de IDE estándar. Los proyectos estándar usan un **IDE-generated Ant build script** para construir y ejecutar su proyecto.
- **Enterprise Application Client with Existing Sources:** Importa una aplicación enterprise cliente existente en un proyecto de IDE estándar. La aplicación enterprise módulo cliente puede tener cualquier estructura. Los proyectos estándar usan un **IDE-generated Ant build script** para construir, ejecutar y depurar su proyecto.
- **Packaged Archive:** Java EE el proyecto del archivo binario. Este tipo de proyecto es un deploytool dónde puede abrir los archivos binarios, editar el servidor runtime deployment descriptors, montar el archivo y utilizar este, para un servidor en blanco Java EE.

3.4.1.1.4 Java ME

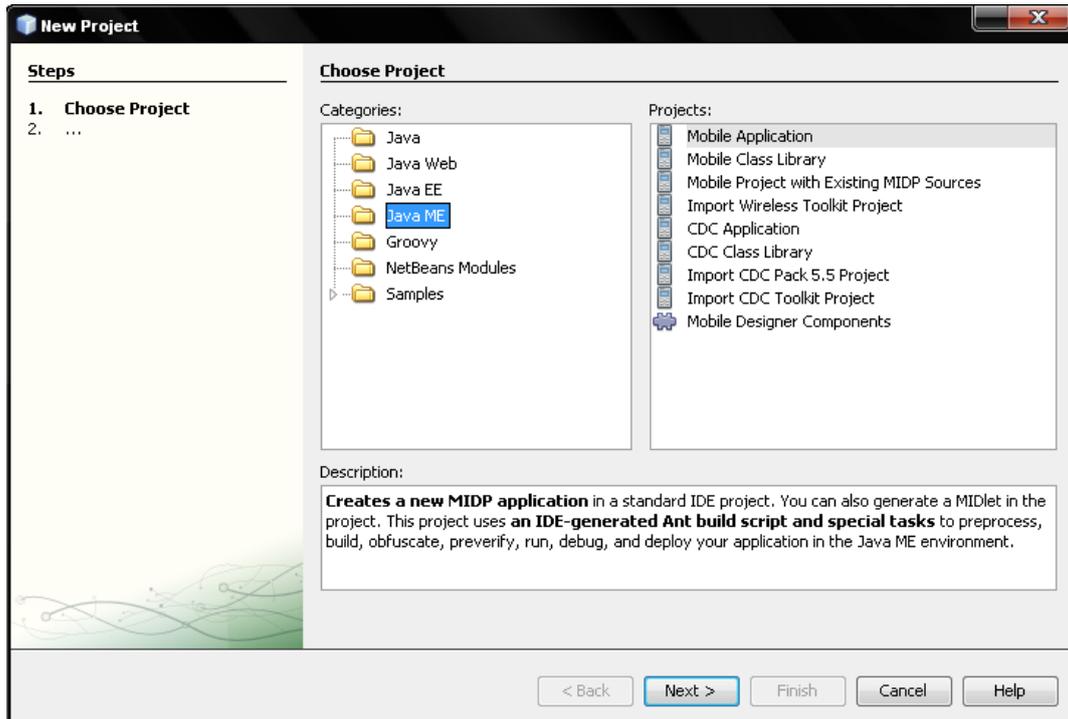


Figura 3. 7. Ventana New Project Java ME

- **Mobile Application:** Crea una nueva aplicación MIDP en un proyecto de IDE estándar. También puede generar un MIDlet en el proyecto. Este proyecto usa un **IDE-generated Ant build script and special tasks** para pre proceso, construir, ofuscar, ejecutar, depurar, y desplegar su aplicación en el ambiente Java ME.
- **Mobile Class Library:** Crea una nueva librería de clase Java ME en un proyecto de IDE estándar. Una librería Java ME no contiene un MIDlet. Este proyecto usa un **IDE-generated Ant build script and special tasks** para pre proceso, construir, ofuscar, ejecutar, depurar, y desplegar su librería en el ambiente Java ME.
- **Mobile Project with Existing MIDP Sources:** Importa una aplicación MIDP existente desde fuentes existentes y una aplicación opcional descriptores en un proyecto de IDE estándar. Este proyecto usa un **IDE-generated Ant build script and special tasks** para pre proceso, construir,

ofuscar, ejecutar, depurar, y desplegar su aplicación en el ambiente Java ME.

- ***Import Wireless Toolkit Project:*** Importa una aplicación MIDP existente del Sun Java Wireless Toolkit en un proyecto de IDE estándar. Este proyecto usa un **IDE-generated Ant build script and special tasks** para pre proceso, construir, ofuscar, ejecutar, depurar, y desplegar su aplicación en el ambiente J2ME.
- ***CDC Application:*** Crea una nueva aplicación CDC en un proyecto de IDE estándar. También puede generar una clase principal en el proyecto. Los proyectos estándar usan un **IDE-generated Ant build script** para construir, ejecutar y depurar su proyecto.
- ***CDC Class Library:*** Crea una nueva librería CDC en un proyecto de IDE estándar. Una librería CDC no contiene una clase principal. Los proyectos estándar usan un **IDE-generated Ant build script** para construir, ejecutar y depurar su proyecto.
- ***Import CDC Pack 5.5 Project:*** Importa un proyecto CDC Pack 5.5 existente en un NetBeans 6.0 proyecto de IDE. . Este proyecto usa un **IDE-generated Ant build script and special tasks** para pre proceso, construir, ofuscar, ejecutar, depurar, y desplegar su aplicación en el ambiente CDC. Se copian los archivos de la fuente del proyecto original.
- ***Import CDC Toolkit Project:*** Importa un proyecto CDC Toolkit existente en un NetBeans 6.0 proyecto de IDE. Este proyecto usa un **IDE-generated Ant build script and special tasks** para pre proceso, construir, ofuscar, ejecutar, depurar, y desplegar su aplicación en el ambiente CDC. Se copian los archivos de la fuente del proyecto original.

- **Mobile Designer Components:** Ayudas para crear el módulo de Componentes de Diseñador Móvil Visual. El Nuevo Módulo de NetBeans se creará con el Diseñador Components Móvil Visual y Java SE Biblioteca Descriptors.

GROOVY

- **Grails Application:** Crea una aplicación nueva y vacía Grails, aplicación para ejecutar "grails crear-app" en el IDE y abriendo la aplicación que es generada como resultado.

NETBEANS MODULES

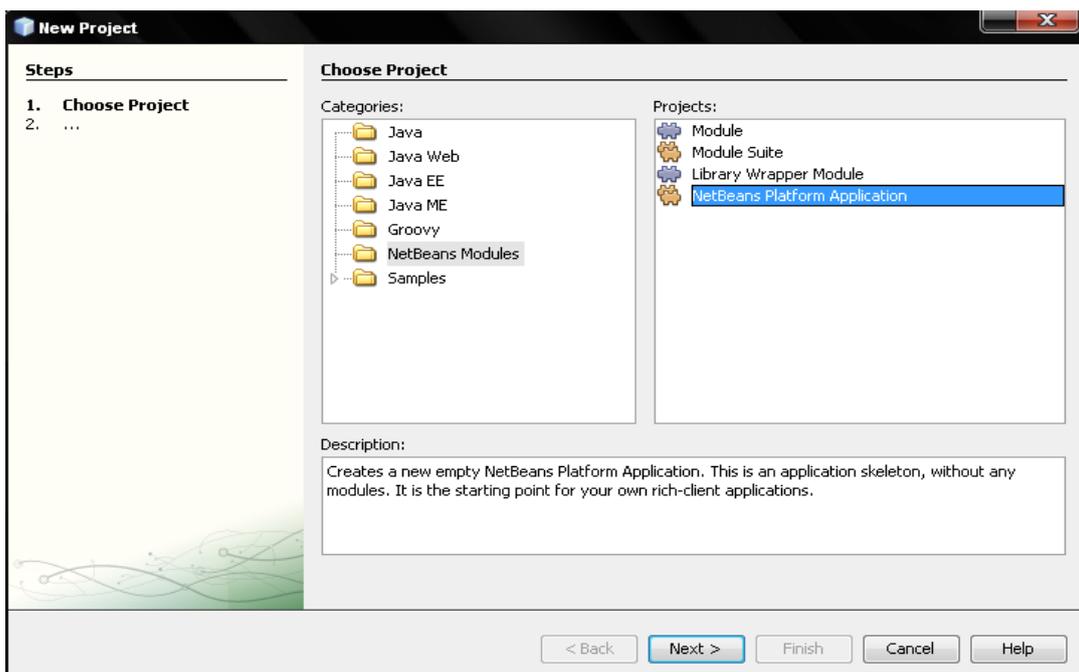


Figura 3. 8. Ventana New Project NetBeans Modules

- **Module:** Crea un nuevo Módulo vacío de NetBeans.
- **Module Suite:** Crea una nueva colección de módulos NetBeans vacía. Esto consiste de un juego de módulos relacionados que pueden construirse juntos.

- **Library Wrapper Module:** Crea un nuevo módulo de NetBeans que es simplemente una envoltura para una librería JAR existente. Otros módulos en la misma colección pueden declarar una dependencia en el módulo de la envoltura a compartirse el acceso a las clases de la biblioteca.
- **NetBeans Platform Application:** Crea una nueva Aplicación Plataforma NetBeans vacía. Éste es un esqueleto de la aplicación, sin ningún módulo. Es el punto de partida para sus propias aplicaciones del rico-cliente.

3.4.1.2 NAME AND LOCATION

En la ventana de Name and Location del asistente, haga lo siguiente (Figura 3.9.):

1. En el campo Project name, escriba HolaMundo.
2. No marque la casilla Use Dedicated Folder for Storing libraries.
3. En el campo Create Main Class escriba, HolaMundo.HolaMundo.
4. Deje marcada la casilla Set as Main Project.

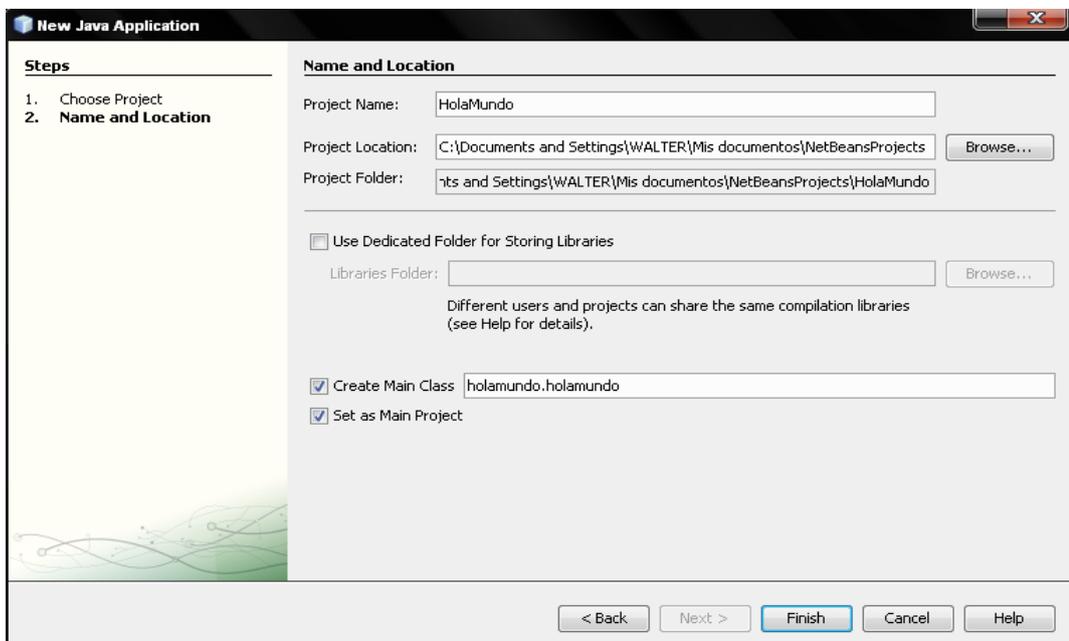


Figura 3. 9. Ventana New Java Application

5. Haga clic en Finish.

El proyecto es creado y está listo para comenzar a trabajar en el IDE. Deberá ver los siguientes componentes:

1. La ventana Projects, que contiene una vista en árbol de los componentes del proyecto, incluyendo el código fuente, las librerías de las cuales se depende, etc.
2. La ventana Source Editor con un archivo abierto llamado HolaMundo.
3. La ventana Navigator, que puede utilizar para moverse rápidamente entre los elementos dentro de la clase seleccionada.

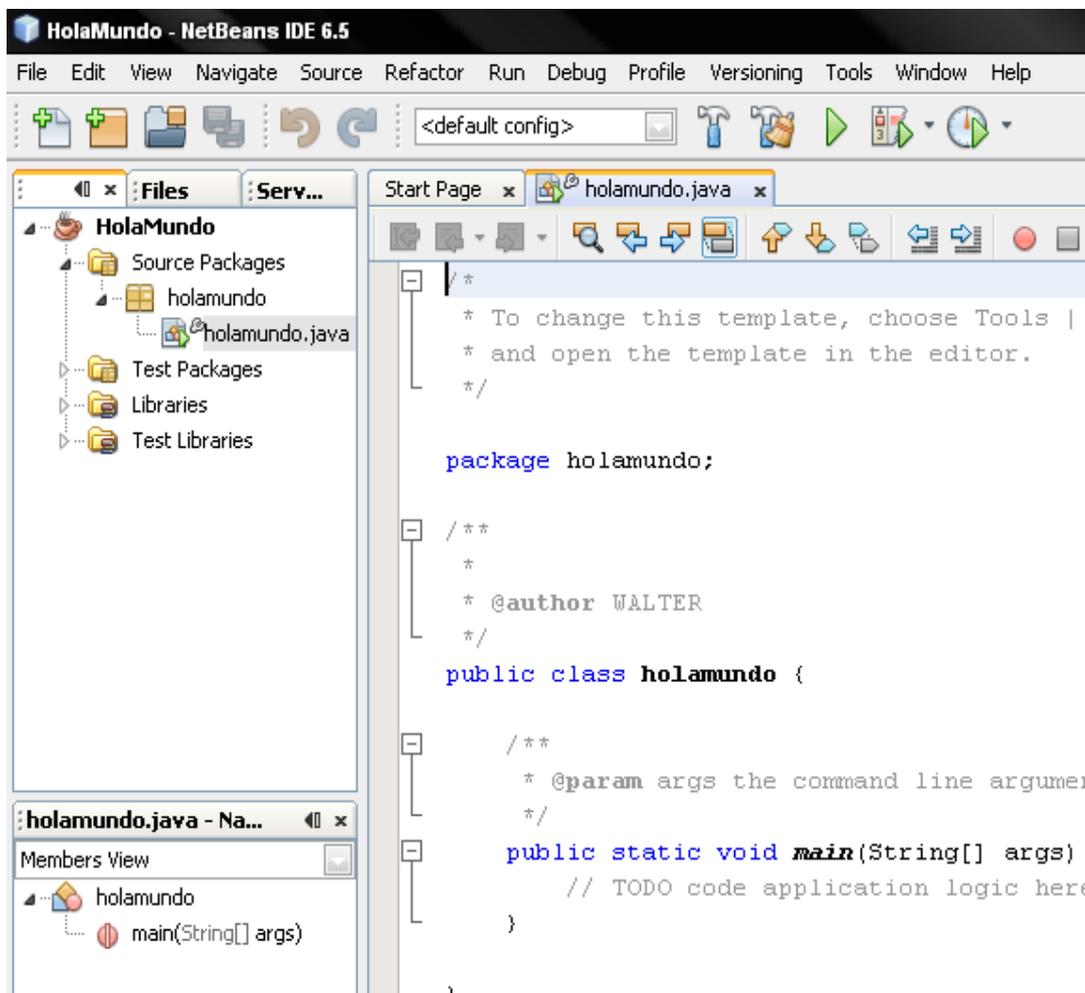


Figura 3. 10. Creación Aplicación Hola Mundo

- **package** holamundo;

- En esta primera línea de código, estamos indicando que la clase holamundo pertenece al paquete holamundo
- **public class** HolaMundo { }
 - Aquí estamos declarando el nombre de la Clase
 - Es muy importante mencionar que todo código Java que se escriba tiene que estar obligatoriamente incluido dentro de una Clase
- **public static void** main(String[] args) { }
 - Aquí estamos declarando el método main(...)
 - De momento saber que la MVJ (Máquina virtual Java) cuando tiene que ejecutar la aplicación holamundo, lo primero que hace es buscar este método como punto de entrada a la aplicación.

Y para que nos muestre el mensaje **Hola Mundo**, tenemos que añadir el siguiente código al método main(...)

```
public static void main(String[] args) {  
    // TODO code application logic here  
    System.out.println("Hola Mundo!");  
}
```

- **System.out.println("Hola Mundo...");**
 - De momento sólo saber que "Hola Mundo" es un String que se muestra como mensaje de salida

3.4.1.3 CREACIÓN DE UN ARCHIVO.

Para crear un archivo utilizando NetBeans 6.5 lo primero que hay que hacer es crear un proyecto. Un proyecto nos permite administrar los archivos con el código fuente y compilado de una aplicación. Para crear un archivo se sigue el siguiente procedimiento:

3.4.1. Del menú principal de NetBeans 6.5, seleccione la opción File/New File ... , presione las teclas Ctrl+N o haga clic en el icono New File mostrado en la figura 3.11.

Aparecerá la primera ventana del asistente para crear un nuevo proyecto, figura 3.12.

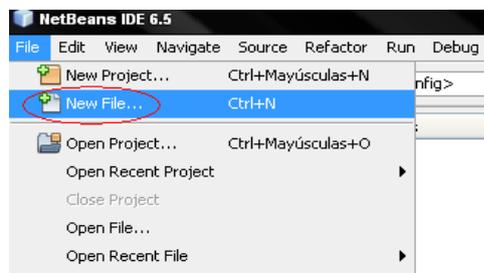


Figura 3. 11. Ventana New File

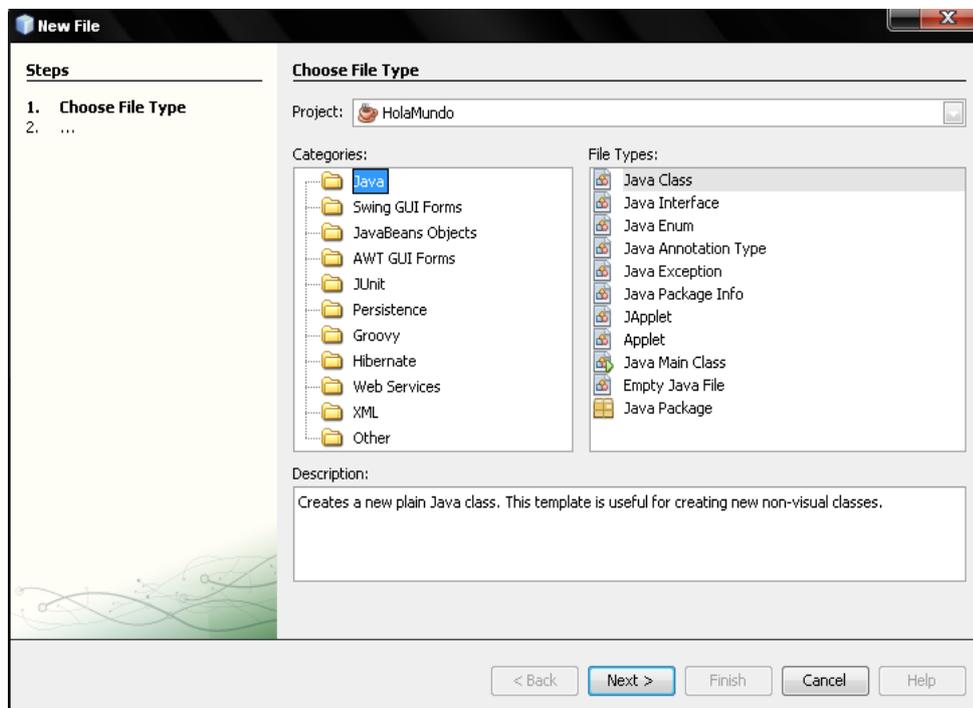


Figura 3. 12. Ventana New File Java Class

En esta ventana del asistente selecciona el tipo de archivo que desea crear y luego presiona el botón **Next>**.

Las categorías que encontramos en la creación de un nuevo archivo, son los siguientes:

JAVA (FIGURA 3.12.)

- **Java Class:** Crea una nueva clase llana en Java. Esta plantilla es útil para crear las nuevas clases non-visuales.
- **Java Interface:** Crea una nueva interfaz de Java.
- **Java Enum:** Crea un nuevo archivo enum Java. Para poder usar el archivo enum, su proyecto debe usar JDK 5.0 o compatible.
- **Java Annotation Type:** Crea un nuevo tipo de anotación Java. Para poder usar los archivos de tipo de anotación, su proyecto debe usar JDK 5.0 o compatible.
- **Java Exception:** Crea una nueva sub clase de Exception con un mensaje detallado opcional.
- **Java Package Info:** Crea un nuevo paquete-info Java. Para poder usar las anotaciones del paquete, su proyecto debe usar JDK 5.0 o compatible.
- **JApplet:** Crea un nuevo applet JFC (el Balance) el applet. Un applet es una clase de Java que puede correr en cualquiera navegador Java-habilitado. **Nota:** Esta plantilla no contiene código del formulario que le permite diseñar el applet visualmente en el Editor del Formulario. Para el

plan visual, empiece con la plantilla de JApplet bajo los Java GUI Formularios

- **Applet:** Crea un nuevo AWT (Ventana Abstracta Toolkit) applet. Un applet es una clase de Java que puede correr en cualquiera navegador Java-habilitado. **Nota:** Esta plantilla no contiene código del formulario que le permite diseñar el applet visualmente en el Editor del Formulario. Para el plan visual, empiece con la plantilla de Applet bajo los Java GUI Formularios | los Formularios d
- **Java Main Class:** Crea una nueva clase de Java con un método principal que le permite ser ejecutado como una aplicación de la consola. Si quiere diseñar una aplicación visual, usted podría preferir usar la plantilla de JFrame bajo los Formularios Java GUI, o un esqueleto de la aplicación bajo los Formularios Java GUI | los Formularios de la Muestra.
- **Empty Java File:** Crea un archivo fuente Java vacío. Ningún código se genera salvo la declaración del paquete requerida. Use esta plantilla para crear una clase desde el principio.
- **Java Package:** Crea un paquete para sostener los archivos fuente Java. Este paquete toma el formulario de una carpeta vacía físicamente en su disco.

SWING GUI FORMS (FIGURA 3.13.)

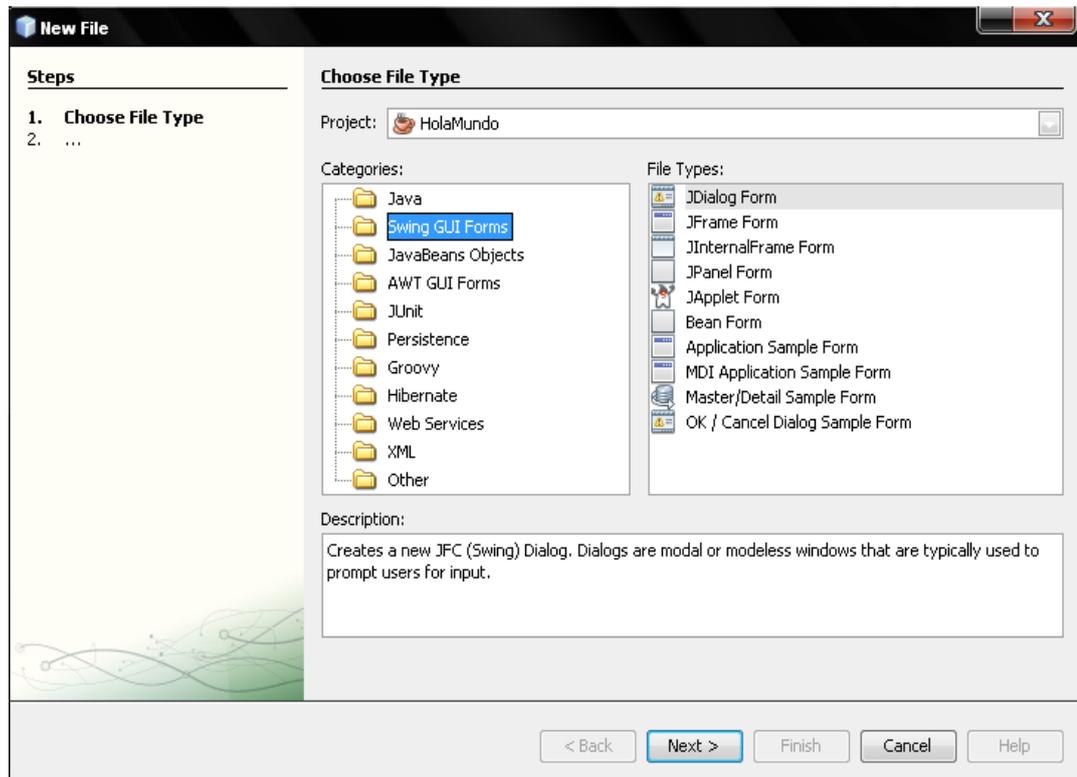


Figura 3. 13. Ventana New File Swing GUI Form

- **JDialog Form:** Crea un nuevo Diálogo JFC (Swing). Los diálogos son modelos de ventanas que se usan típicamente a los usuarios puntuales para la entrada.
- **JFrame Form:** Crea un nuevo Frame JFC (Swing). Frames se usan típicamente como ventanas top-level, como la principal interfaz del usuario a la aplicación. Más aplicaciones Swing se construyen empezando desde este formulario.
- **JInternalFrame Form:** Crea un nuevo frame interior JFC (el Balance) que puede ponerse hacia un JDesktopPane para proporcionar al usuario una interfaz MDI. Típicamente usted crearía una clase del frame interior que contiene los elementos del despliegue para cada frame individual. Trate cada clase del frame interior como una clase de JFrame. Entonces cree una aplicación JFrame (o JApplet) con un JDesktopPane en él. Agregue los

frame interiores a la hoja del panel del escritorio que usa el código, o usando las funciones de Copiar y Pegar en la ventana de los Proyectos después de que los frames interiores son compilados.

- **JPanel Form:** Crea un nuevo panel JFC (Swing). Los Paneles son frecuentemente usados para agrupar varios componentes en un lugar. Un JPanel puede usarse como un componente dentro de un JFrame o JDialog.
- **JApplet Form:** Crea un nuevo Applet JFC (Swing). Un applet es una clase Java que puede ejecutarse en cualquiera navegador Java-habilitado.
- **Bean Form:** Crea un nuevo formulario basado en cualquier componente de JavaBeans (visual o no-visual). Usted debe especificar la súper-clase del formulario bean en el asistente.
- **Application Sample Form:** Crea una esquema aplicación JFrame-base, incluyendo tres menús.
- **MDI Application Sample Form:** Crea una aplicación JFrame con los ítems del menú común pre-definidos y conteniendo un JDesktopPane que los frames interiores pueden agregarse al runtime. Esta plantilla puede usarse como una ventana principal para las aplicaciones MDI.
- **Master/Detail Sample Form:** Crea un nuevo formulario de Master/Detail.
- **OK/Cancel Dialog Sample Form:** Crea un diálogo Swing pre-construido con botones OK y Cancelar. Usted puede usar esta plantilla para crear el mensaje en cajas y cualquier otro diálogo. **TIP:** Usted también puede crear otras plantillas para los diálogos. (Por ejemplo, usted podría querer una plantilla con botones Sí, No, y Cancelar.) Primero cree el diálogo y luego haga clic derecho en el nodo del archivo fuente en la ventana de Proyectos y escoge Guardar como Plantilla del menú contextual.

JAVABEANS OBJECTS (FIGURA 3.14.)

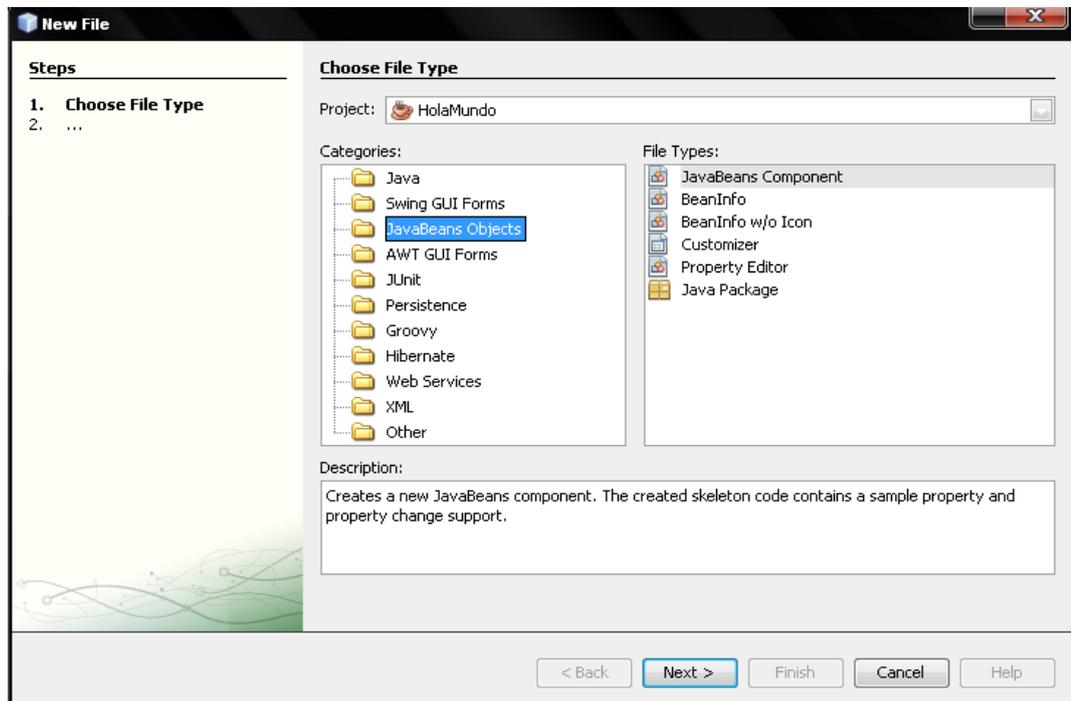


Figura 3. 14. Ventana New File JavaBeans Objects

- **JavaBeans Component:** Crea un nuevo componente JavaBeans. El código esquema creado contiene una propiedad de muestra y una propiedad de soporte de cambio.
- **BeanInfo:** Crea una nueva clase BeanInfo. El BeanInfo se usa como una descripción de cómo su componente de JavaBeans asociado debe comportarse y debe aparecer.
- **BeanInfo w/o Icon:** Crea una nueva clase de BeanInfo. El BeanInfo se usa como una descripción de cómo su componente de JavaBeans asociado debe comportarse y debe aparecer.
- **Customizer:** Crea un panel Personalizado para un componente JavaBeans. El personalizar puede usarse para configura el bean en el IDE.

- **Property Editor:** Crea una clase PropertyEditor. Los editores de propiedad se usan en diseños IDEs para soportar correcciones de las propiedades de componentes JavaBeans.
- **Java ackage:** Crea un paquete para sostener los archivos fuente Java. Este paquete toma el formulario de una carpeta vacía físicamente en su disco.

AWT GUI FORMS (FIGURA 3.15.)

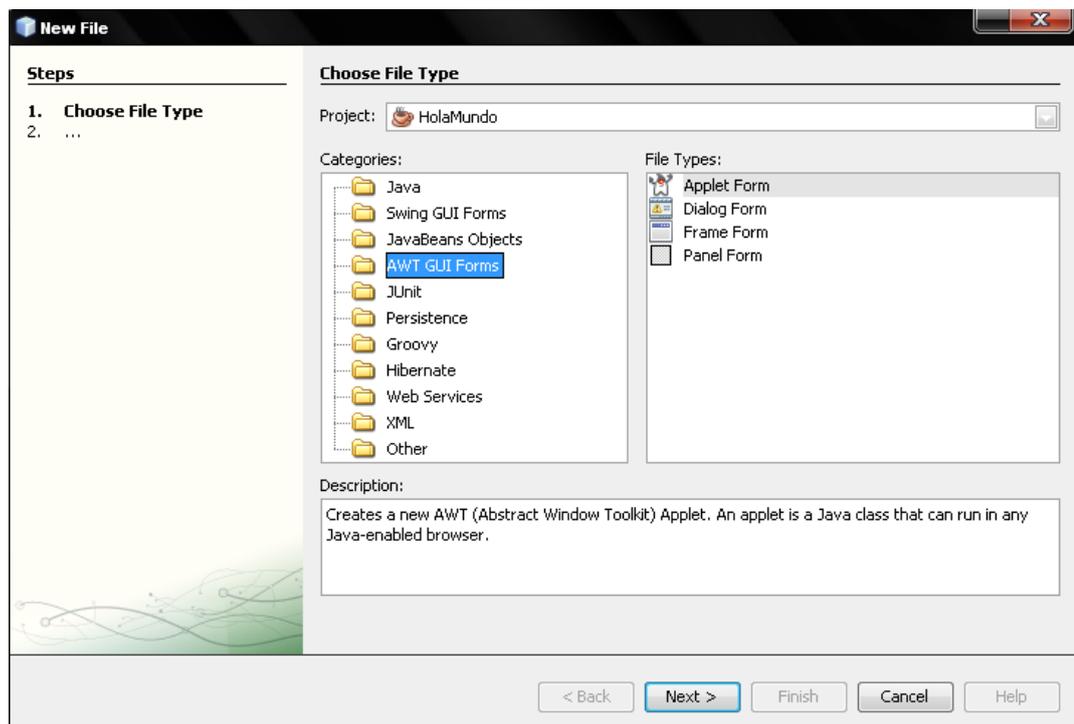


Figura 3. 15. Ventana New File AWT GUI Forms

- **Applet Form:** Crea un nuevo applet AWT (Ventana Abstracta Toolkit). Un applet es una clase Java que puede ejecutarse en cualquiera navegador Java-habilitado.
- **Dialog Form:** Crea un nuevo formulario de dialogo AWT (la Ventana Abstracta Toolkit). Los formularios de diálogos son modelos de ventanas que usan típicamente los usuarios puntuales para la entrada.

- **Frame Form:** Crea un nuevo Frame AWT (Ventana Abstracta Toolkit). Los frames se usan típicamente como ventanas top-level, como la principal interfaz del usuario a la aplicación.
- **Panel Form:** Usando esta plantilla, usted puede diseñar un nuevo panel AWT (Ventana Abstracta Toolkit).

Los paneles son frecuentemente se usados para agrupan varios componentes en un lugar. Un panel puede usarse como un componente dentro de un Frame o Diálogo.

JAVABEANS OBJECTS (FIGURA 3.16.)

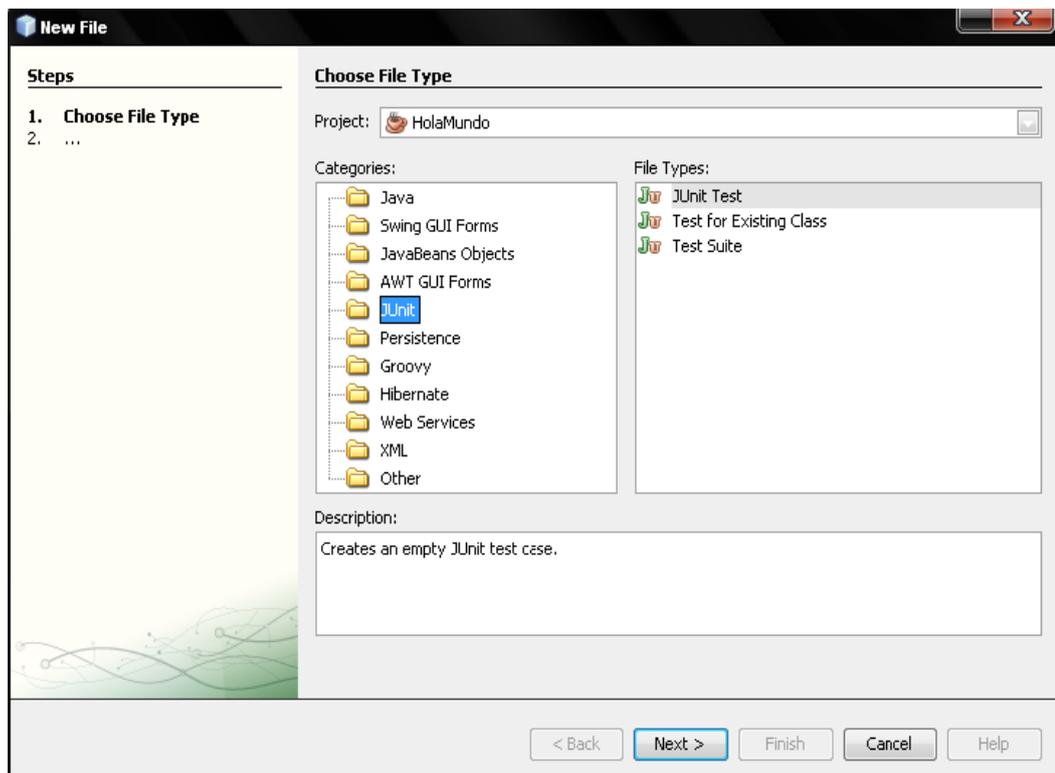


Figura 3. 16. Ventana New File JUnit

- **JUnit Test:** Crea una prueba de caso JUnit vacío.

- **Test for Existing Class:** Crea una prueba de caso JUnit simple para probar métodos de una sola clase.
- **Test Suite:** Crea una colección de prueba para todas las pruebas en un paquete Java seleccionado.

PERSISTENCE (FIGURA 3.17.)

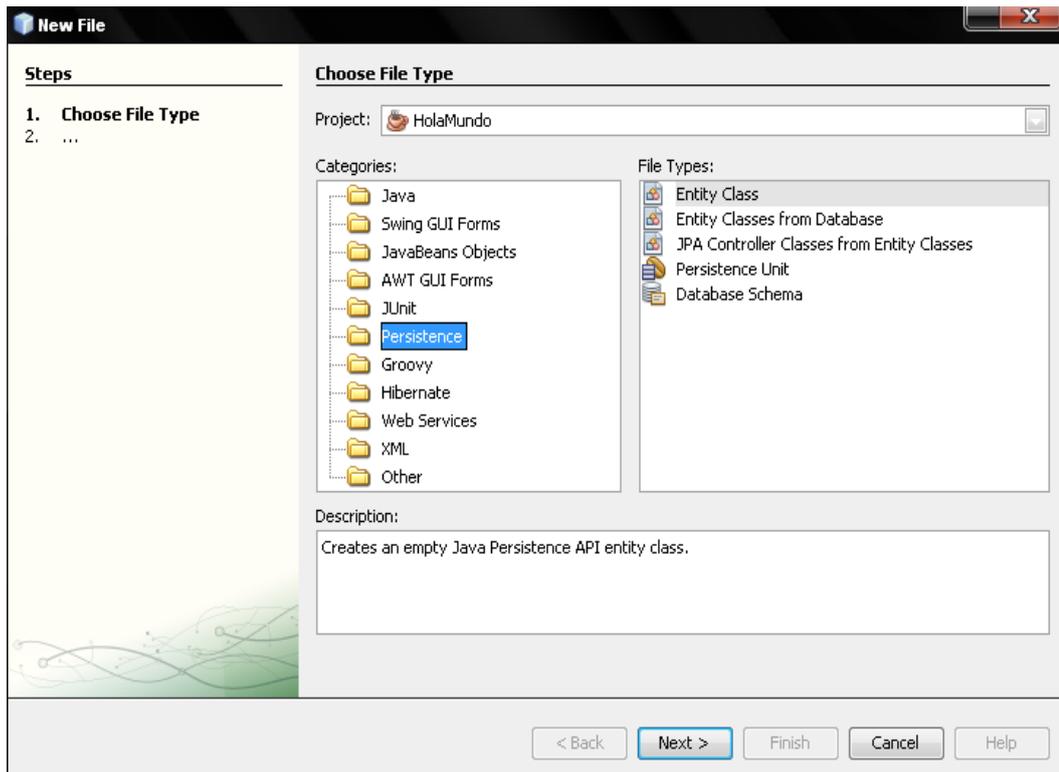


Figura 3. 17. Ventana New File Persistence

- **Entity Class:** Crea una entidad clase Java Persistencia API vacía.
- **Entity Classes from Database:** Crea una entidad clases Java Persistencia API basadas en una base de datos relacional existente. Se usan las clases de la entidad para representar objetos cuyo lifespan son más largos que una ejecución del programa típica. Esta plantilla crea una clase entidad para cada tabla seleccionada, campos que representan las columnas, y relaciones que representan las claves primarias.

- **JPA Controller Classes from Entity Classes:** Crea un set JPA controlador de clases y clases relacionadas desde un set de clases entidad.
- **Persistence Unit:** Crea una unidad persistencia JSR220. Si persistence.xml no existe, se crea.
- **Database Schema:** Crea un nuevo archivo de esquema persistente de base de datos.

GROOVY

- **Groovy Class:** Crea una nueva clase groovy. Groovy es un idioma del scripting.
- **Groovy Script:** Crea una nueva escritura groovy. Groovy es un idioma del scripting.

HIBERNATE (FIGURA 3.18.)

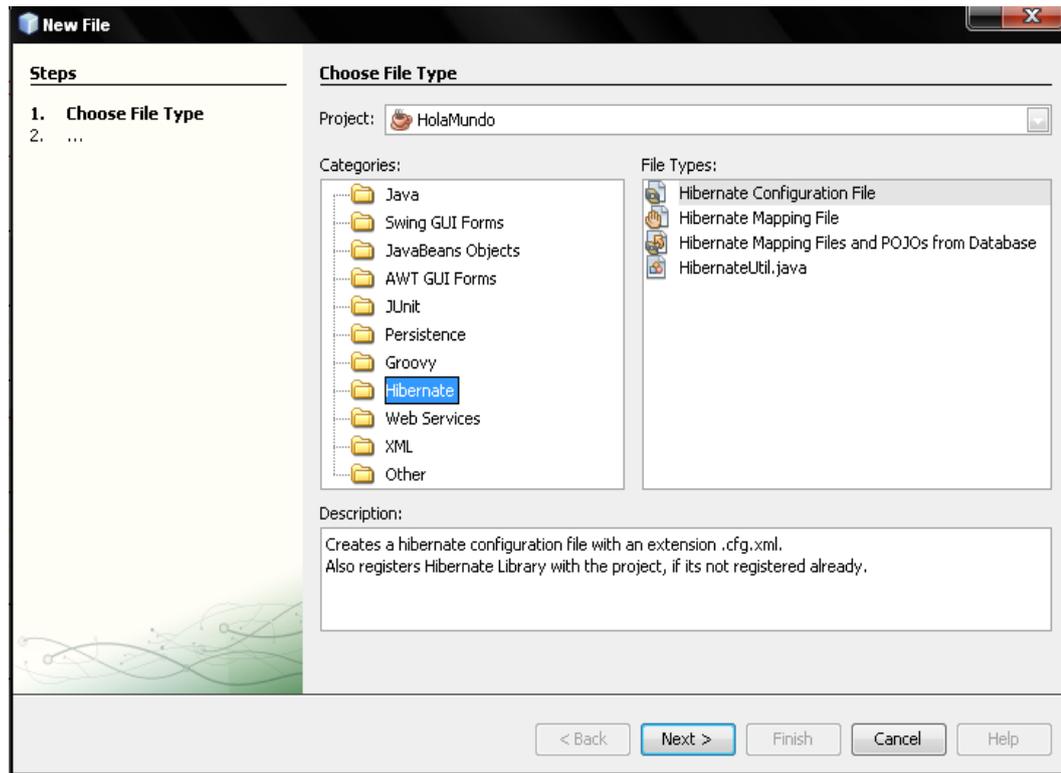


Figura 3. 18. Ventana New File Hibernate

- **Hibernate Configuration File:** Crea un archivo de configuración hibernate con una extensión. cfg.xml. También registros de Librería Hibernate, con el proyecto, si este ya no se registró.
- **Hibernate Mapping File:** Crea un archivo trazado hibernate, con una extensión. hbm.xml.
- **Hibernate Mapping Files and POJOs from Database:** Crea un archivo trazado Hibernate, y POJOs basaron en una base de datos relacional existente. Este asistente también crea un archivo de ingeniería reservado con una extensión. reveng.xml.
- **HibernateUtil.java:** Crea un clase de utilidad hibérnate que habilita la recuperación fácil de la Sesión de Objetos Hibernate.

- **Web Services**

- **Web Service Client:** Crea a un cliente de servicio web que es flexible con JSR-109.
- **Logical Handler:** Crea a un gestor lógico simple. Esta clase formula que usted personalice para satisfacer sus necesidades.
- **Message Handler:** Crea a un gestor de mensajes SOAP simple. El gestor de mensajes SOAP intercepta el mensaje SOAP como éste hace desde el cliente para el servicio y vice-versa. La clase generada realiza un registro simple para solicitar su mensaje SOAP. Esta clase formula que usted personalice para satisfacer sus necesidades.

XML (FIGURA 3.19.)

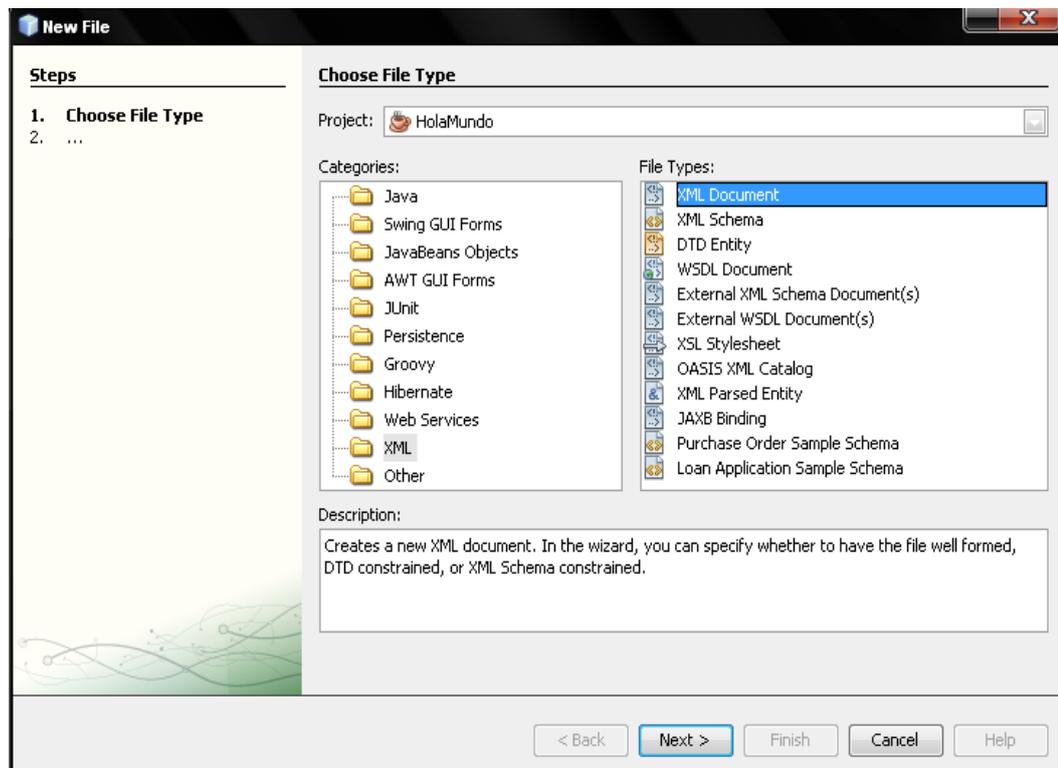


Figura 3. 19. Ventana New File XML

- **XML Document:** Crea un nuevo documento de XML. En el asistente, usted puede especificar si tiene bien el formato de archivo, reprimió DTD, o reprimió el Esquema de XML.
- **XML Schema:** Crea un Esquema documento XML que especifica los namespace y tipos de datos para un documento XML.
- **DTD Entity:** Crea una nueva definición de tipo de documento (DTD). DTDs describen las etiquetas que pueden usarse en un archivo XML y pueden indicarse el arreglo válido de esas etiquetas.
- **WSDL Document:** Crea un nuevo archivo WSDL con un esquema XML incluido o importado.
- **External XML Schema Document(s):** Este Asistente recupera documentos Esquema y sus dependencias.
- **External WSDL Document(s):** Este Asistente recupera documentos WSDL y sus dependencias.
- **XSL Stylesheet:** Crea un documento hoja de estilo extensible (XSL) vacía. Usa documentos XSL para especificar las características del despliegue, las conversiones de la etiqueta, y envió de mecanismos para su documento XML.
- **OASIS XML Catalog:** Crea un archivo Catalogo OASIS XML que traza las referencias de la entidad externas a los recursos locales. Usted puede agregar el catálogo para un DTD y Esquema Catálogos XML en la ventana de Runtime.
- **XML Parsed Entity:** Crea un objeto entidad analizado externo.
- **JAXB Binding:** Crea clases Java para un Esquema dado.
- **Purchase Order Sample Schema:** Crea un ejemplo de documento esquema PurchaseOrder.xsd.

- **Loan Application Sample Schema:** Crea un ejemplo de documento esquema LoanApplication.xsd.

OTHER (FIGURA 3.20.)

En esta categoría se crea otros tipos de archivos:

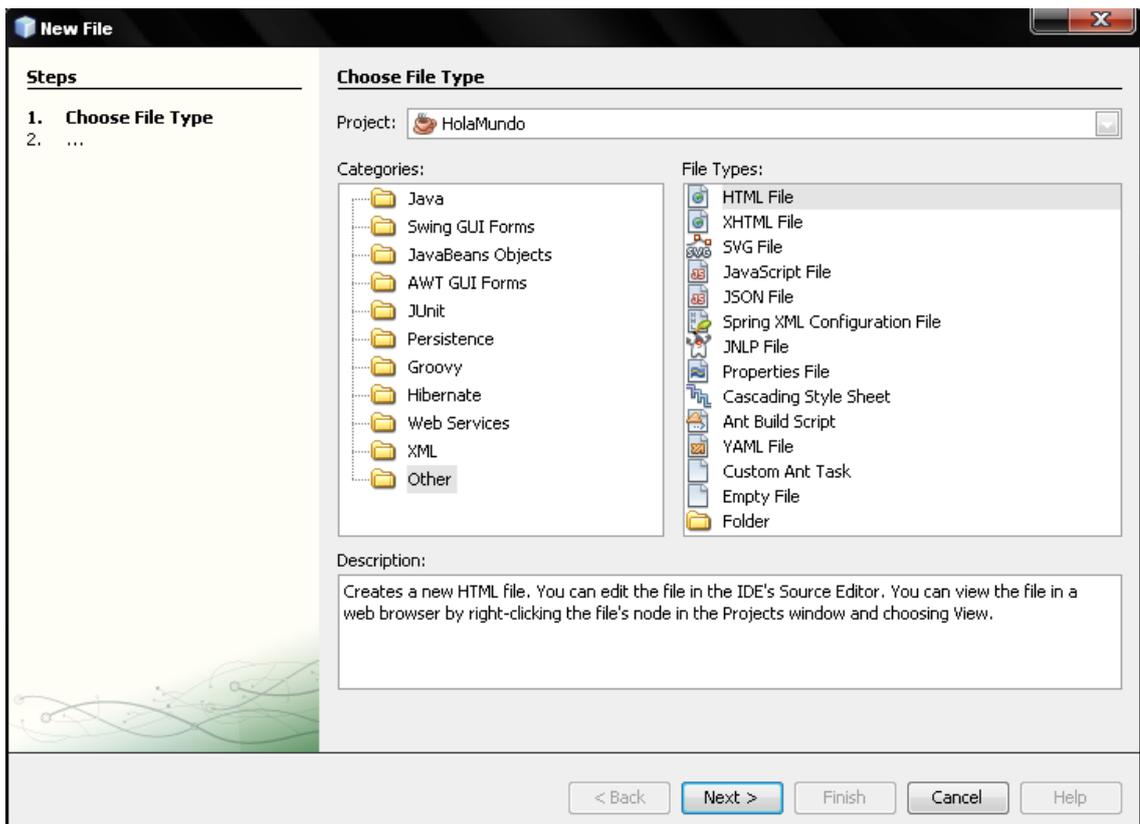


Figura 3. 20. New file de Otros Tipos de aplicaciones

- **HTML File:** Crea un nuevo archivo HTML. Usted puede revisar el archivo en el Editor de la Fuente del IDE. Usted puede ver el archivo en un navegador web.
- **XHTML File:** Crea un nuevo archivo Estricto XHTML. Usted puede revisar el archivo en el Editor de la Fuente del IDE.

- **SVG File:** Crea un archivo de Gráfico del Vector Escalable (SVG) vacío Tiny 1.1. Usted puede revisar el archivo en el Editor de la Fuente del IDE o puede lanzar a un editor externo para hacer los cambios al archivo.
- **JavaScript File:** Crea el archivo de JavaScript vacío. Usted puede revisar el archivo en el Editor de la Fuente del IDE. Y usted puede ejecutarlo usando el menú de contexto de editor. Para cambiar esta plantilla, escoja las Herramientas | Gerente de la Plantilla y abre la plantilla en el editor.
- **JSON File:** Crea el archivo JSON vacío. JSON (Notación de Objeto JavaScript) es un formato de datos-intercambio ligero, el subconjunto de JavaScript. Es fácil para los humanos leer y escribir, y ampliamente usar con aplicaciones de Ájax.
- **Spring XML Configuration File:** Crea un nuevo archivo de configuración Spring XML.
- **Cascading Style Sheet:** Crea un documento de hoja de estilo cascada vacía (CSS). Use un CSS para estructurar la información contenida en su documento XML.
- **Ant Build Script:** Construye un script vacío.
- **YAML File:** Crea un archivo YAML vacío. Usted puede revisar el archivo en el Editor de la Fuente del IDE. Para cambiar esta plantilla, escoja las Herramientas | Gerente de la Plantilla y abre la plantilla en el editor.
- **Custom Ant Task:** Una tarea ANT personalizada.
- **Empty File:** Crea un archivo vacío con una extensión arbitraria en su disco.
- **Folder:** Crea una nueva carpeta en su disco.

3.4.2. Aparecerá en la ventana de Name and Location del asistente, en el campo Class Name, escriba **Hola**, el cual va a hacer el nombre de la clase (archivo) dentro del proyecto **HolaMundo**.

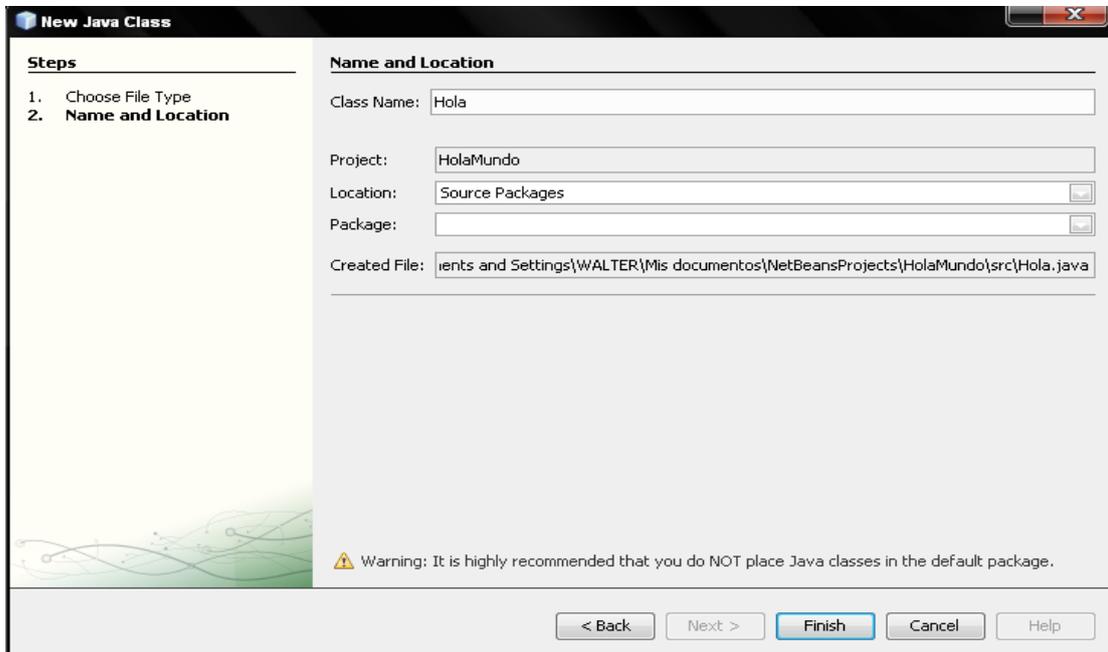


Figura 3. 21. Ventana New Java Class

- Haga clic en Finish

El archivo es creado y está listo para comenzar a trabajar en el IDE.

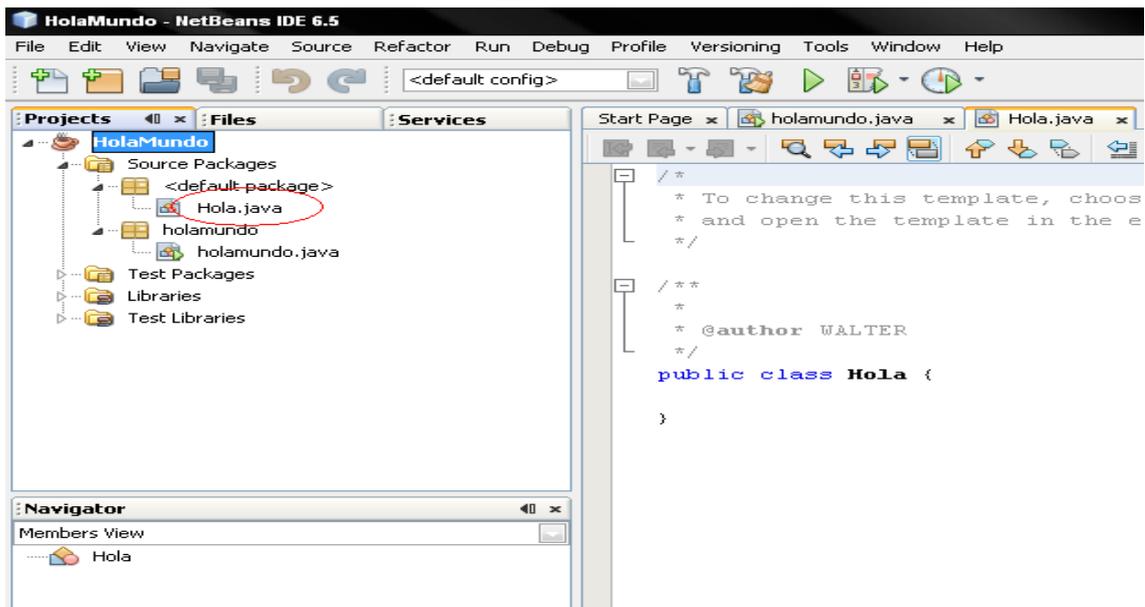


Figura 3. 22. Ventana de la aplicación Hola Mundo

3.4.1.4 CREAR UN NUEVO GRUPO DE PROYECTO

El grupo de proyecto le permite crear agrupaciones de proyectos para que usted pueda abrir y puede cerrar varios proyectos en seguida. Además, cada grupo puede tener un proyecto principal diferente.

Del menú principal de NetBeans 6.5, seleccione la opción **File/New File ...** , clic en el icono **Project Group > New Group** mostrado en la figura 3.23.

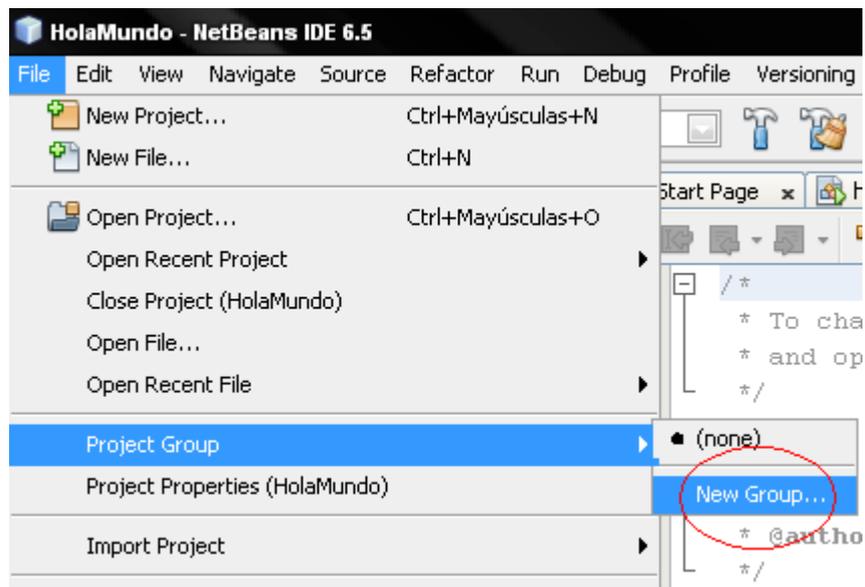


Figura 3. 23. Ventana Project Group

Usted puede crear los tipos siguientes de grupos:

- **Grupo libre.** Cuando usted crea un grupo libre, usted puede escoger ambas opciones de las siguientes:
- **Use los Proyectos Actualmente Abiertos.** Agrega los proyectos actualmente abiertos al grupo.
- **Automáticamente Excepto la Lista del Proyecto.** Los cambios los volúmenes del grupo del proyecto según los proyectos que usted abre y cierra cuando el grupo del proyecto está abierto.

El proyecto y todos los Proyectos Requeridos. Crea un grupo que consiste en un proyecto maestro y todos los proyectos que de este depende.

La carpeta de Proyectos. Crea un grupo fuera de todos los proyectos que están dentro de una carpeta que usted especifique.

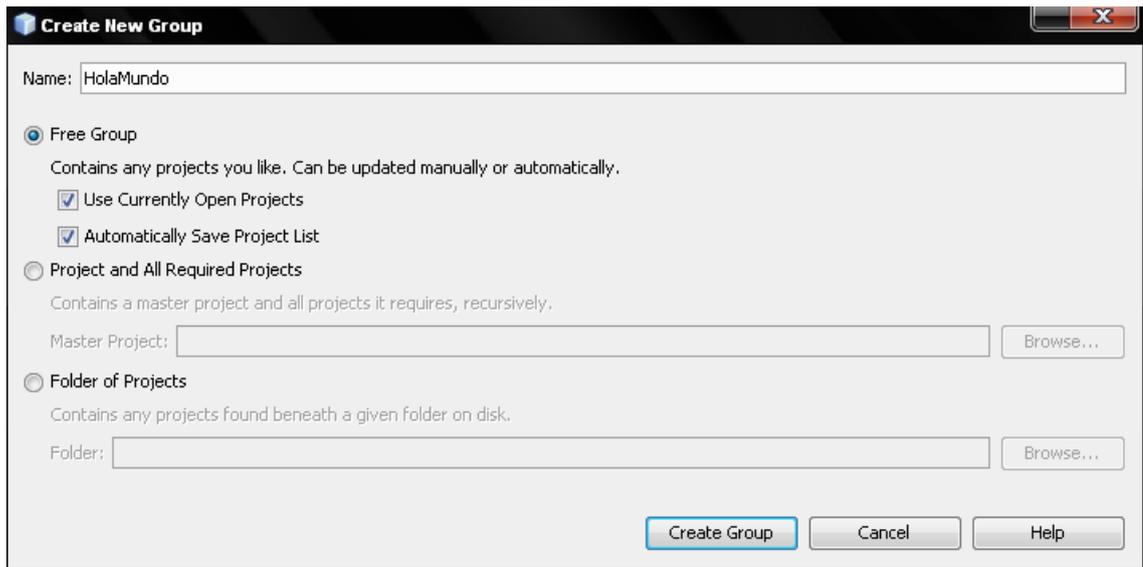


Figura 3. 24. Ventana Create New Group

Una vez que ha creado un grupo de proyecto, ese grupo de proyecto está disponible en el menú **File > Project Group**. Cuando selecciona un grupo del proyecto de ese menú, todos los otros proyectos abiertos se cierran. Si quiere cerrar todos los proyectos abiertos, escoge del menú **File > Project Group > (ninguno)**.

Si quiere crear un grupo que contiene un juego específico de proyectos, se asegura que esos proyectos son los únicos proyectos abiertos. Entonces escoja **File > Project Group > New Group**. En el cuadro de diálogo de Nuevo Grupo, seleccione el botón de **grupo libre** y seleccione la casilla de verificación de los **Proyectos Actualmente Abierta**.

Si no quiere los volúmenes de un grupo libre para cambiar, abra y cierre los proyectos, elimine automáticamente y guarde la Lista de Proyecto.

Si quiere modificar los volúmenes de un grupo libre automáticamente seleccione la opción de Lista de Proyecto, abra cualquier proyecto que quiere agregar y cierre cualquier proyecto que quiere quitar del grupo. Entonces escoja el **File > Project Group > Properties of GroupName** y hace clic ahora guarde la Lista del Proyecto Actual.

PROPIEDADES DEL PROYECTO

Las Propiedades del Proyecto podemos encontrar escogiendo **File ><el nombre del proyecto activo> propiedades**, esto abre el cuadro de diálogo de Propiedades de Proyecto.

En este cuadro de dialogo podemos personalizar las propiedades del proyecto, para esta configuración escogemos las propiedades del árbol de menú en el panel izquierdo del cuadro de diálogo de Propiedades.

Escoja agregar la Configuración desde el menú desplegable para agregar una nueva configuración al proyecto.

Haga clic en el botón Manage Configurations para abrir un cuadro de diálogo que despliega la lista completa de configuraciones creado para el proyecto seleccionado. En el cuadro de dialogo Project Configuration Manager, usted puede agregar, puede reproducir, o puede quitar las configuraciones del proyecto, o crear una plantilla de configuración de proyecto.

3.4.2 MENÚ EDIT

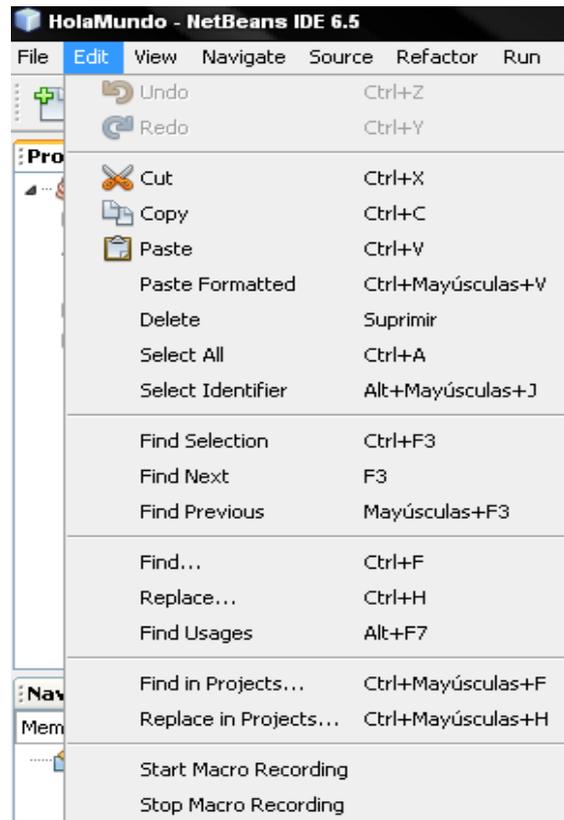


Figura 3. 25. Ventana MENÚ EDIT

3.4.2.1 Comando Undo (Ctrl-Z)

Descripción

Marcha atrás (uno a la vez) una serie de acciones del editor, excepto guardar

3.4.2.2 Comando Redo (Ctrl-Y)

Descripción

Marcha atrás (uno a la vez) una serie de órdenes del comando Undo.

3.4.2.3 Comando Cut (Ctrl-X)

Descripción

Suprime la selección actual y la coloca en el portapapeles.

3.4.2.4 Comando Copy (Ctrl-C)

Descripción

Copia la selección actual al él portapapeles.

3.4.2.5 Comando Paste(Ctrl-V)

Descripción

Pega el contenido del portapapeles en el punto de inserción.

3.4.2.6 Comando Paste Formatted (Ctrl-Mayúsculas-V)

Descripción

Pega formatos contenidos en el portapapeles en el punto de la inserción.

3.4.2.7 Comado Delete

Descripción

Borra la selección actual.

3.4.2.8 Comando Select All (Ctrl-A)

Descripción

Selecciona todo en el documento actual o ventana.

3.4.2.9 Comando Find Selection (Ctrl-F3)

Descripción

Busca instancias (ejemplos) de la selección actual.

3.4.2.10 Comando Find Next (F3)

Descripción

Busca las siguientes instancias (ejemplos) encontradas en el texto.

3.4.2.11 Comando Find Previous (Mayúsculas-F3)

Descripción

Busca las anteriores instancias (ejemplos) encontradas en el texto.

3.4.2.12 Comando Find (Ctrl-F)

Descripción

Busca una cadena de texto.

3.4.2.13 Comando Replace (Ctrl-H)

Descripción

Busca una cadena de texto y esta es reemplazada con una cadena específica.

3.4.2.14 Comando Find Usages (Alt-F7)

Descripción

Busca usos y subtipos de código seleccionado.

3.4.2.15 Comando Find in Project (Ctrl-Mayúsculas-F)

Descripción

Busca texto específico, nombres de objetos, tipos de objetos dentro de los proyectos.

3.4.2.16 Comando Replace in Projects (Ctrl-Mayúsculas-H)

Descripción

Reemplaza texto, nombres de objetos, tipos de objetos dentro de los proyectos.

3.4.2.17 Comando Start Macro Recording

Descripción

Inicia la grabación de un macro que contiene pulsaciones y movimientos del cursor.

3.4.2.18 Comando Stop Macro Recording

Descripción

Para la grabación de macro.

3.4.3 MENÚ VIEW

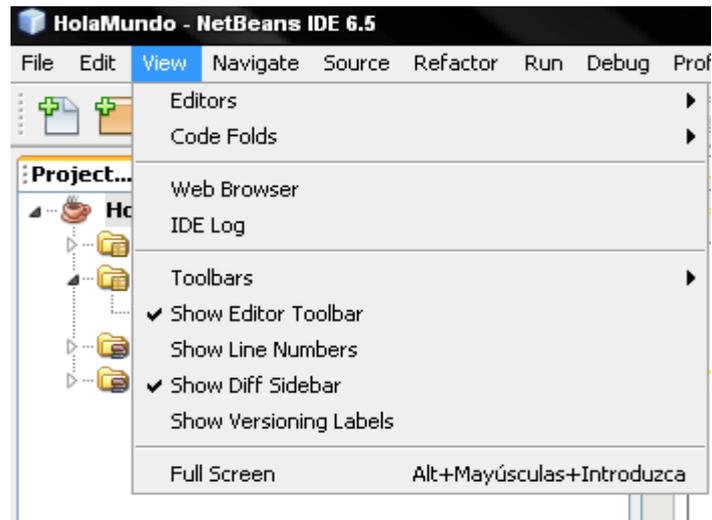


Figura 3. 26. Ventana MENÚ VIEW

3.4.3.1 Comando Code Folds (Pliegues del código)

3.4.3.1.1 Collapse Fold (Ctrl-Numpad - -)

Si el punto de inserción está en una sección plegable de texto, estas se desplazan línea a línea.

3.4.3.1.2 Expand Fold (Ctrl-Numpad - +)

Si la línea seleccionada actualmente en el Editor de la Fuente es representada por algunos pliegues de líneas, extiende el pliegue para mostrar todas las líneas.

3.4.3.1.3 Collapse All (Ctrl-cambio-Numpad - -)

Despliega todas las secciones plegables de texto en el Editor de la Fuente.

3.4.3.1.4 Expand All (Ctrl-cambio-Numpad - +)

Expande todas las secciones plegables de texto en el Editor de la Fuente.

3.4.3.1.5 Full Screen (Alt-Mayúsculas-Enter)

Expande la ventana completa en su longitud y ancho de pantalla.

3.4.3.2 Toolbars

El IDE proporciona el mando comprensivo de cómo se presentan los toolbars. Puede agregar, posicionar y controlar la visibilidad del toolbars. Puede crear las configuraciones del toolbar también personalizadas.

Escoja **View > Toolbars** del menú principal para ver una lista del toolbars disponible. El submenú despliega todo el toolbars en el IDE.

3.4.3.2.1 Para agregar un toolbar a la ventana principal:

- De la ventana principal, escoja la **View > Toolbars > Customize**
- En la caja de dialogo de Customize Toolbars, pulse el botón Nuevo Toolbar.
- En la Nueva Toolbar diálogo caja, teclee un nombre para el toolbar y clic OK.

3.4.3.2.2 El IDE agrega un toolbar vacío a la ventana principal.

- Mostrar o esconder un toolbar:

Toolbars que son actualmente visibles son indicados por una marca de control en el Toolbars el menú automático.

- Escoja la **View > Toolbars** del menú principal y escoge un nombre del toolbar en el submenú.

Alternadamente, pulse clic derecho en el botón vacío del toolbar y escoge el nombre del toolbar desde el menú automático.

3.4.3.2.3 Para mover un toolbar:

Haga clic en el lado izquierdo de la textura del toolbars y arrastre el toolbar a la situación deseada.

3.4.4 MENÚ NAVIGATE

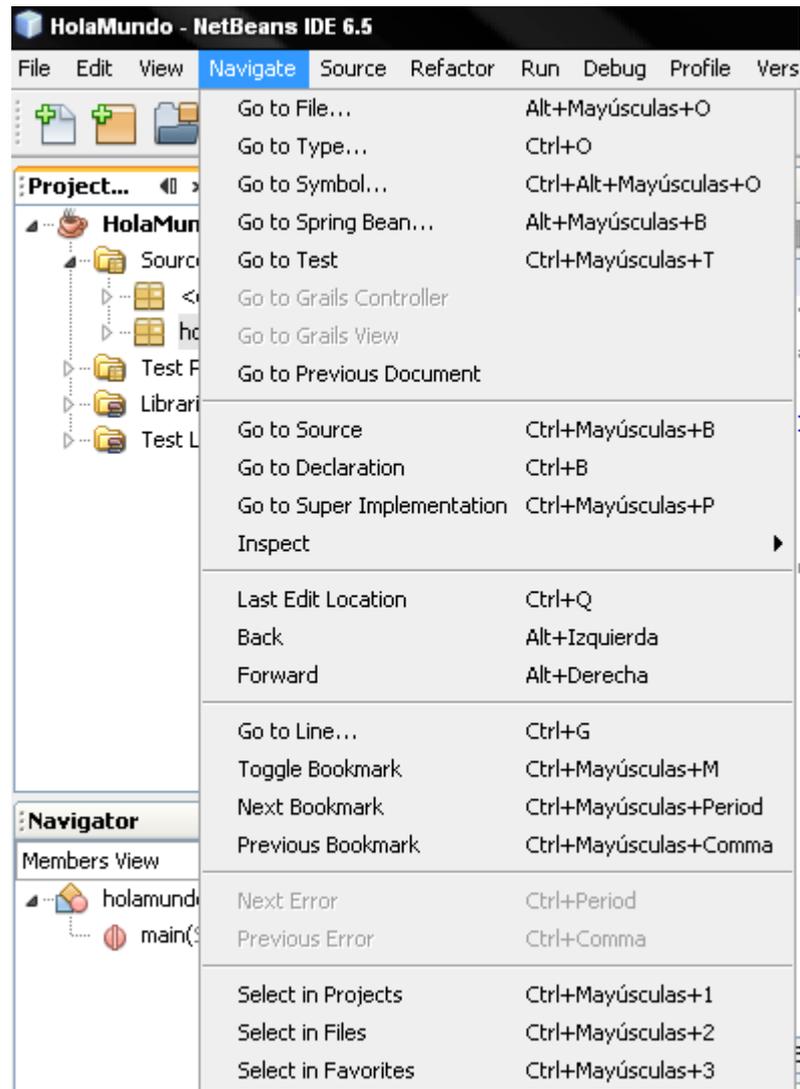


Figura 3. 27. Ventana NAVIGATE

3.4.4.1 Comando Go to File (Alt-Mayúsculas-O)

Descripción

Busca y abre un archivo específico.

3.4.4.2 Comando Go to Type (Ctrl-O)

Descripción

Busca y abre una clase específica o interfaz.

3.4.4.3 Comando Go to Test (Ctrl-cambio-T)

Descripción

Busca y abre una prueba específica.

3.4.4.4 Comando Go to Previous Document

Descripción

Abre el último documento abierto antes del actual.

3.4.4.5 Comando Go to Source (Ctrl-Mayúsculas-B)

Descripción

Despliegue la fuente del archivo que contiene la definición de la clase seleccionada.

3.4.4.6 Comando Go to Declaration (Ctrl-B)

Descripción

Salte a la declaración del artículo bajo el cursor.

3.4.4.7 Comando Go to Super Implementation (Ctrl-Mayúsculas-P)

Descripción

Salte a la súper implementación del artículo bajo el cursor.

3.4.4.8 Comando Last Edit Location (Ctrl-Q)

Descripción

Desplace al editor al último lugar dónde ocurrió la corrección.

3.4.4.9 Comando Back (Alt-Left)

Descripción

Navegue atrás

3.4.4.10 Comando Forward (Alt-Right)

Descripción

Navegue adelante

3.4.4.11 Comando Go to Line (Ctrl-G)

Salte a la línea especificada.

3.4.4.12 Comando Toggle Bookmark (Ctrl-Mayúsculas-M)

Ponga un marcador de libros en una línea de código.

3.4.4.13 Comando *Next Bookmark* (Ctrl-Mayúsculas-Period)

Ciclo adelante a través de los marcadores de libros.

3.4.4.14 *Previous Bookmark* (Ctrl-Mayúsculas-Comma)

Ciclo al revés a través de los marcadores de libros.

3.4.4.15 Next Error (Ctrl-Period)

Desplaza el Editor de la Fuente a la próxima línea que contiene el error en la estructura.

3.4.4.16 Previous Error (Ctrl-Comma)

Desplaza el Editor de la Fuente a la línea anterior que contenía el error en la estructura.

3.4.4.17 Select in Project (Ctrl-Mayúsculas-1)

Abre la ventana de Proyectos y selecciona el documento actual dentro de él.

3.4.4.18 Select in Files (Ctrl-Mayúsculas-2)

Abre la ventana de Archivos y selecciona el documento actual dentro de él.

3.4.4.19 Select in Favorites (Ctrl-Mayúsculas-3)

Abre la ventana de Favoritos y selecciona el documento actual dentro de él.

3.4.5 MENÚ SOURCE

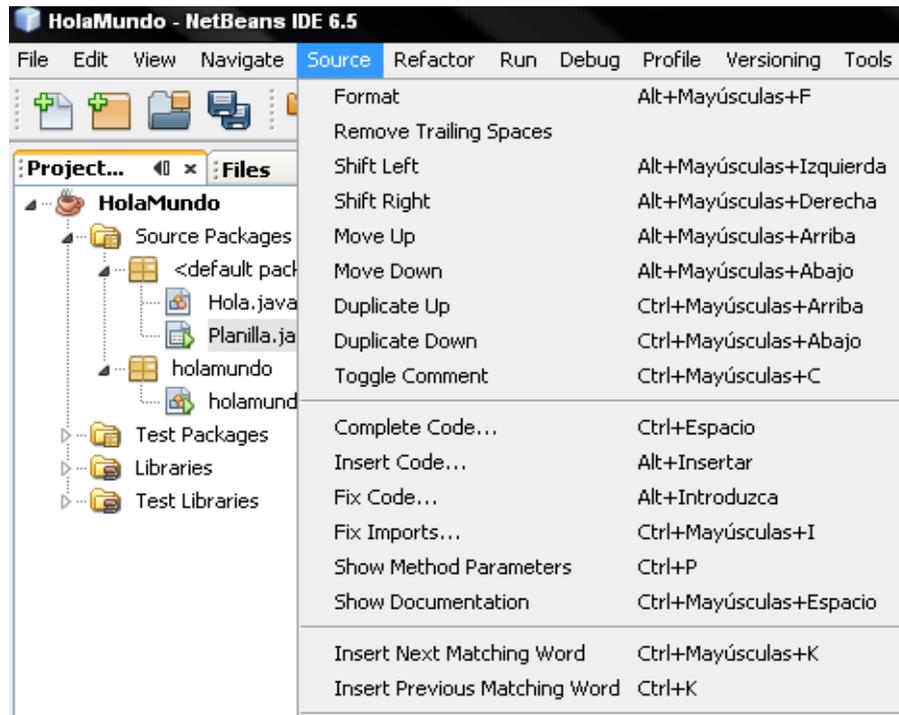


Figura 3. 28. Ventana MENÚ SOURCE

3.4.5.1 Format (Alt-Mayúsculas-F)

Formatea el código seleccionado o el archivo entero si nada es seleccionado.

3.4.5.2 Shift Left (Alt-Mayúsculas-Left)

Mueve la línea o líneas seleccionadas a la izquierda.

3.4.5.3 Shift Right (Alt-Mayúsculas-Right)

Mueve la línea o líneas seleccionadas al derecho.

3.4.5.4 Move Up (Alt-Mayúsculas-Up)

Mueve la línea o líneas seleccionadas una línea arriba.

3.4.5.5 *Move Down* (Alt-Mayúsculas-Down)

Mueve la línea o líneas seleccionadas una línea abajo.

3.4.5.6 *Duplicate Up* (Ctrl-Mayúsculas-Up)

Copie la línea o líneas seleccionadas una línea arriba.

3.4.5.7 *Duplicate Down* (Ctrl-Mayúsculas-Down)

Copie la línea o líneas seleccionadas una línea abajo.

3.4.5.8 *Toggle Comment* (Ctrl-Mayusculas-C)

Las barras Slash para comentario fuera de la línea o líneas seleccionadas.

3.4.5.9 *Complete Code* (Ctrl-Space)

Muestra la caja de completación de código.

3.4.5.10 *Insert Code* (Alt-Insert)

Poner un menú consciente de contexto que usted puede usar para generar las estructuras comunes como constructores, getters, y setter.

3.4.5.11 *Fix Code* (Alt-Enter)

Despliega el editor de sugerencias. El IDE lo informa cuando una sugerencia está disponible y cuando se despliega.

3.4.5.12 Fix Imports (Ctrl-Shift-I)

Genera las declaraciones de import requeridas por las clases especificadas.

3.4.5.13 Show Method Parameters (Ctrl-P)

Selecciona el próximo parámetro. Usted debe tener un parámetro seleccionado (resaltó) para trabajar con un acceso directo.

3.4.5.14 Show Documentation (Ctrl-Shift-Space)

Muestre la documentación del artículo bajo el cursor.

3.4.5.15 Insert Next Matching Word (Ctrl-Shift-K)

Genera la siguiente palabra usada en otra parte de su código como usted teclee sus caracteres al principio.

3.4.5.16 Insert Previous Matching Word (Ctrl-K)

Genera la palabra anterior usada, en otra parte de su código como usted teclee sus caracteres del principio.

3.4.6 MENÚ RUN

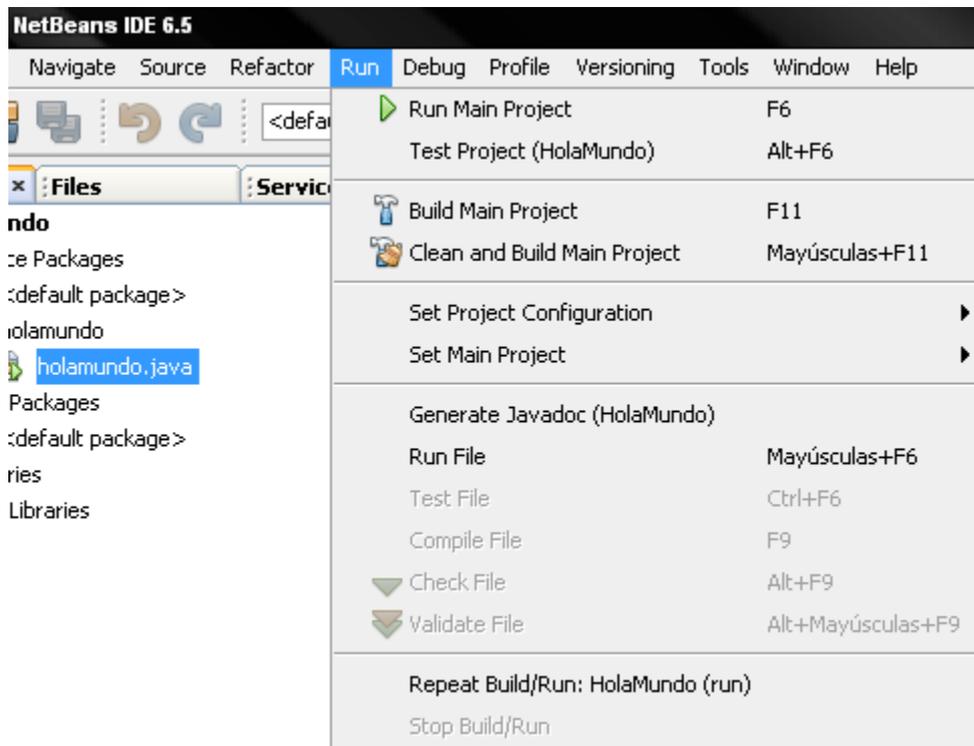


Figura 3. 29. Ventana MENÚ RUN

3.4.6.1 Run Main Project (F6)

Ejecuta el proyecto principal.

3.4.6.2 Debug Main Project (Ctrl-F5)

Depura el proyecto principal.

3.4.6.3 Test Project (Alt-F6)

Inicia la prueba del proyecto JUnit.

3.4.6.4 Run File (Shift-F6)

Ejecuta el archivo actualmente seleccionado.

3.4.6.5 Debug File (Ctrl-Shift-F5)

Inicia sesión de depuración del archivo actualmente seleccionado.

3.4.6.6 Test File (Ctrl-F6)

Inicia prueba del archivo actual JUnit.

3.4.6.7 Debug Test for File (Ctrl-Shift-F6)

Inicia prueba de depuración para el archivo en JUnit.

3.4.6.8 Finish Debugger Session (Shift-F5)

Finaliza la sesión de depuración.

3.4.6.9 Continue (F5)

Resumen de depuración hasta el próximo punto de ruptura o el fin del programa es alcanzado.

3.4.6.10 Step Over (F8)

Ejecuta una línea de la fuente de un programa. Si la línea es una llamada de método, ejecuta el método entero luego para.

3.4.6.11 Step Over Expression (Shift-F8)

Pasa por encima de la expresión y entonces detiene la depuración.

3.4.6.12 Step Into (F7)

Ejecuta una línea de la fuente de un programa. Si la línea es una llamada de método, ejecuta el programa, primero las declaraciones del método y para.

3.4.6.13 Step Out (Ctrl-F7)

Ejecuta una línea de la fuente de un programa. Si la línea es una llamada del método, ejecuta los métodos y retorna el control de la llamada.

3.4.6.14 Run to Cursor (F4)

Ejecuta el proyecto actual a la localización del cursor en el archivo y detiene la ejecución del programa.

3.4.6.15 Run into Method (Shift-F7)

Ejecuta el proyecto actual al método especificado y luego los pasos en el método.

3.4.6.16 Make Callee Current (Ctrl-Alt-Up)

Hace llamar al método inicial de la llamada actual. Sólo disponible cuando una llamada se selecciona en la ventana de Pila de Llamada.

3.4.6.17 Make Caller Current (Ctrl-Alt-Down)

Crea el método llamando a la actual llamada. Sólo disponible cuando una llamada se selecciona en la ventana de Pila de Llamada.

3.4.6.18 Toggle Line Breakpoint (Ctrl-F8)

Agrega una línea a un punto de ruptura o remueve el punto de ruptura a la localización del cursor en el programa.

3.4.6.19 New Breakpoint (Ctrl-Shift-F8)

Coloca un nuevo punto de ruptura a la línea, excepción, o método específico.

3.4.6.20 New Watch (Ctrl-Shift-F7)

Agrega variable especificada para mirar.

3.4.6.21 Evaluate Expression (Ctrl-F9)

Abre el cuadro de dialogo de Evalúe Expresión.

3.4.7 MENÚ DEBUG

Inicio de una sesión de depuración local

Es el proceso de depuración de un programa que se ejecuta en el mismo equipo que el IDE. El IDE inicia el depurador, a continuación, ejecuta la aplicación dentro del depurador. Al iniciar una sesión de depuración, el IDE automáticamente abre las ventanas y el depurador imprime automáticamente en la ventana de rendimiento los resultados.

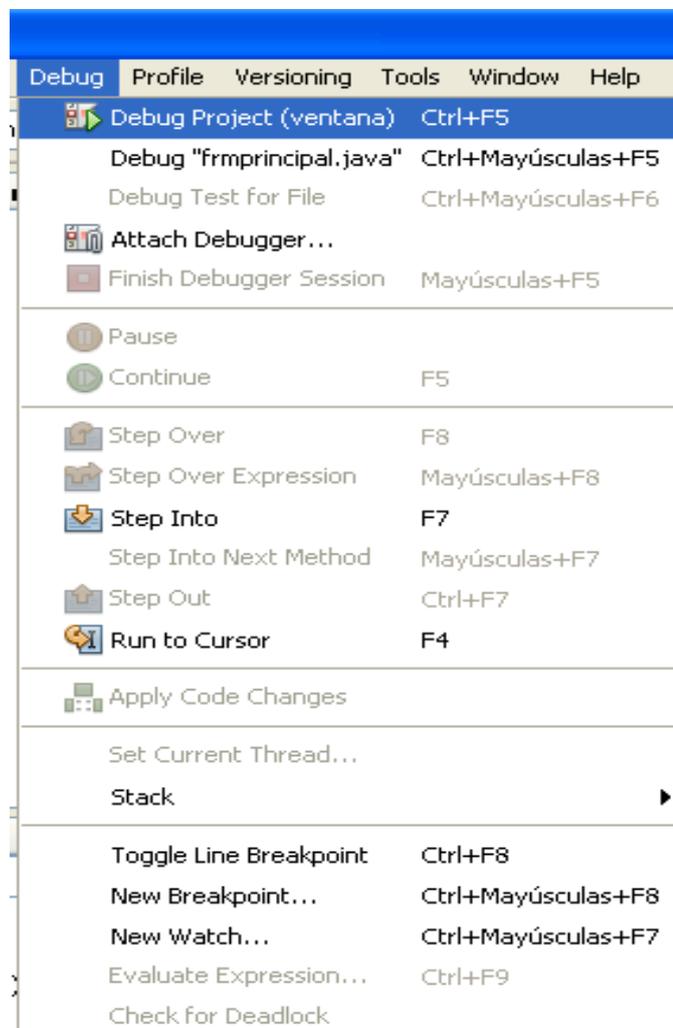


Figura 3. 30. Ventana MENÚ DEBUG

El principal proyecto de depuración

Todos los comandos de depuración en el menú Depurar se ejecutan en el proyecto principal. No importa qué archivo o proyecto seleccionado en la ventana o Proyectos Fuente: Editor.

Estos comandos iniciar la sesión de depuración en la clase principal del proyecto principal.

3.4.7.1 Comando Debug > Debug Main Project (Ctrl + F5)

Descripción

Ejecuta el programa hasta que llega a un punto de interrupción o excepción o hasta que el programa termina normalmente.

3.4.7.2 Comando Attach Debugger

Descripción

Empezando una Sesión de la Depuración Remota

La depuración remota es el proceso de poner a punto una aplicación que está corriendo en una computadora diferente. Esta técnica es útil cuando usted está desarrollando una aplicación que corre en un servidor de tejido o en un ambiente diferente que la computadora en que usted está desarrollando la aplicación.

3.4.7.3 Comando Step Into (F7 o)

Descripción

Ejecuta el programa a la primera línea después de la rutina principal y se detiene la ejecución antes de que se hayan realizado cambios en el estado del programa.

3.4.7.4 Comando Finish Debugger Session (Shift+F5)

Descripción

Si es necesario, usted puede detener la sesión. Usted también puede cerrar una sesión de la depuración especificando que la ventana de Sesión en la que esta.

3.4.7.5 Comandos Pause y Continuar

3.4.7.5.1 Comando Pause .- Detiene la Sesión



Figura 3. 31. Ventana MENÚ DEBUG Pause

3.4.7.5.2 Comando Continuar.- Reanuda la Sesión

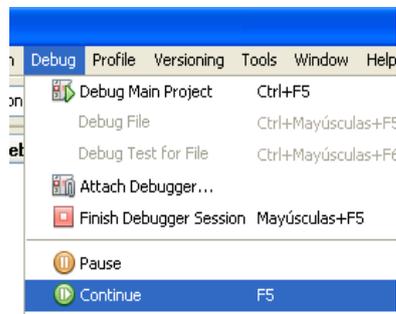


Figura 3. 32. Ventana MENÚ DEBUG Continue

3.4.7.6 Comandos



Figura 3. 33. Ventana de Comandos

3.4.7.7 Step Over (F8)

Descripción

Ejecuta una línea de la fuente. Si la línea de la fuente contiene una llamada, ejecuta la rutina entera sin caminar a través de las instrucciones individuales

3.4.7.8 Step Over Expression (Shift+F8)

Descripción

Ejecuta una llamada de método en una expresión. Si una expresión tiene múltiples llamadas a los métodos, puede utilizar el paso a paso a través de una expresión y ver el valor de cada llamada de método en la expresión en la ventana Variables locales. Cada vez que utilice el comando Paso Más de expresión, el depurador avanza a la siguiente llamada de método en la expresión y el método de llamada se completará subrayado.

3.4.7.8.1 Step Into Next Method(Shift+F8)

Descripción

Ejecuta una línea de la fuente. Si la línea de la fuente contiene una llamada, el IDE simplemente detiene antes de ejecutar la primera declaración de la rutina. Usted también puede empezar una sesión de la depuración con el Paso En el orden. Las paradas de ejecución de programa en la primera línea después de que la rutina principal antes de cualquier cambio se ha hecho al estado del programa.

3.4.7.8.2 Step Out (Ctrl+F7)

Descripción

Ejecuta una línea de la fuente. Si la línea de la fuente es parte de una rutina, ejecuta las líneas restantes de la rutina y mando de los ingresos a la visita de la rutina. La llamada del método completada se resalta en el Editor de la Fuente.

3.4.7.8.3 Comando Run to Cursor (F4 o)

Descripción

Ejecuta el programa a la posición del cursor en el Editor de Origen y se detiene el programa. El archivo que ha seleccionado en el Editor de Fuente debe ser llamado a partir de la clase principal del proyecto principal.

3.4.7.8.4 Debugging an Individual Project

Haga clic derecho en el proyecto en la ventana y elegir los proyectos de depuración.

El IDE se ejecuta el proyecto en ejecución hasta que el depurador se detiene o se llegue a un punto de interrupción.

- **Depurar un archivo individual (Debugging an Individual File)**

Seleccione cualquier archivo ejecutable en la ventana de Proyectos y elegir Debug > Debug my_file.

El IDE se ejecuta el archivo en ejecución hasta que el depurador se detiene o se llegue a un punto de interrupción.

3.4.8 MENÚ PROFILE

Inicio de una reunión de perfiles

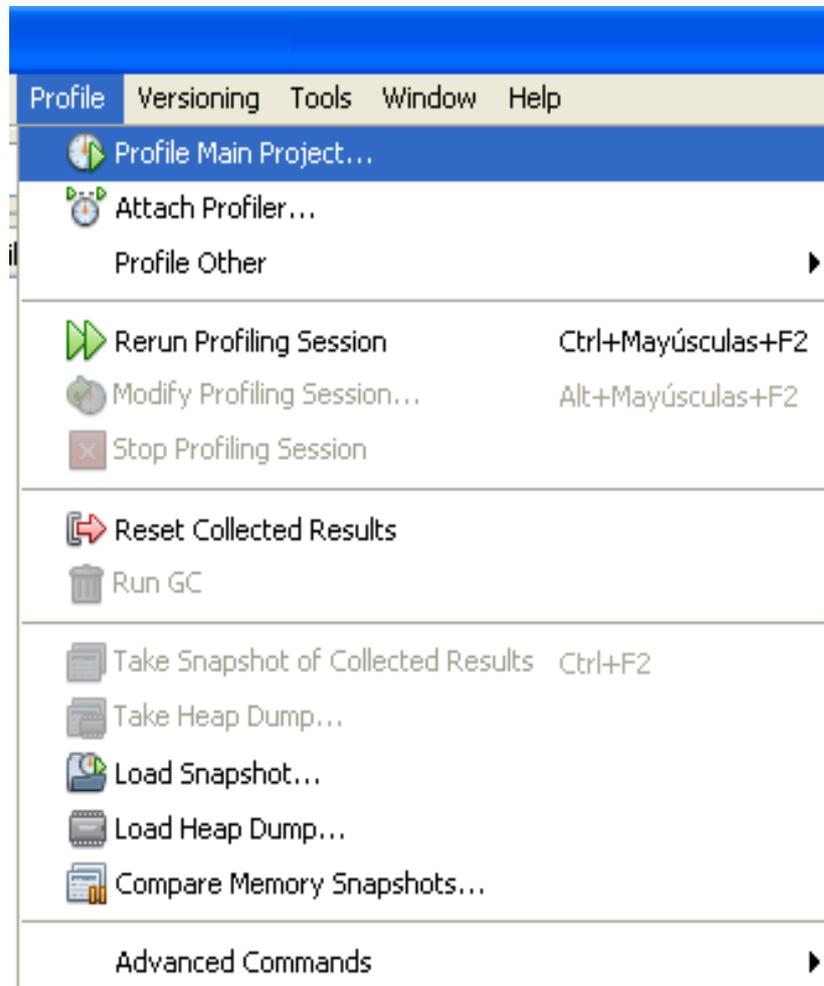


Figura 3. 34. Ventana MENÚ PROFILE

Si usted tiene un proyecto que está dirigido a ejecutar en el equipo local, usted puede perfilar el proyecto sin cualquier configuración adicional. Cuando usted perfila un proyecto local, usted lanza el proyecto y empieza la sesión perfilando desde dentro el IDE.

Si desea un perfil de aplicación local, pero no puede o no quiere iniciar la aplicación desde el IDE, usted puede perfilar de la solicitud adjuntando la IDE a la solicitud.

Para el perfil de un proyecto local, el proyecto debe estar abierto en el IDE. Cada uno de los siguientes perfiles de comandos en el menú Perfil de invocar un cuadro de diálogo donde se selecciona el perfil de tareas para el período de sesiones. Puede iniciar el período de sesiones de perfiles después de seleccionar los perfiles de tarea.

3.4.8.1 Comando Profile > Profile Main Project (Perfil> Perfil del Proyecto Principal)

Descripción

Abre el cuadro de diálogo Perfiles de Tarea. La tarea seleccionada se ejecuta en el proyecto principal.

3.4.8.2 Comando Profile > Attach Profiler (Perfil> Adjuntar Perfil)

Descripción

Abre el cuadro de diálogo Adjuntar Perfil. En el cuadro de diálogo Adjuntar Perfil, seleccionar el proyecto, la tarea de perfiles y el modo de enganche a la solicitud. Utilice este perfil para aplicaciones remotas, correr aplicaciones locales, o las aplicaciones que se inicie fuera del IDE.

3.4.8.3 Comando Profile > Profile Other > "*projectname*"

Descripción

Abre el cuadro de diálogo Perfiles de Tarea. La tarea seleccionada se ejecuta en el proyecto seleccionado.

Notas:

Antes de poder utilizar el profiler en el IDE, es necesario calibrar el profiler. Sólo es necesario para calibrar el perfil de una vez.

La primera vez que el perfil de un proyecto, el IDE tiene que modificar el script proyecto para integrar el profiler. El IDE crea una copia de seguridad de su proyecto antes de script que realiza las modificaciones.

3.4.8.3.1 Para el perfil principal del proyecto

1. Seleccione Profile > Profile Main Project desde el menú principal.
2. Seleccione un profiling task en el cuadro de diálogo Perfiles de Tarea.
3. Especifique las opciones para la tarea.
4. Haga clic en Ejecutar.

Cuando haga clic en Ejecutar, el IDE se inicia el proyecto principal y abre la ventana de Perfil.

3.4.8.3.2 Para el perfil de un proyecto individual

Haga clic derecho en un proyecto en la ventana de Proyectos y elija Perfil de Proyecto.

1. También puede seleccionar un archivo en el proyecto y elija Profile > Profile Other > Profile Project "*projectname*" from the main MENÚ.
2. Seleccione un profiling task en el cuadro de diálogo Perfiles de Tarea.
3. Especifique las opciones para la tarea.
4. Haga clic en Ejecutar.

PARA EL PERFIL DE UNA CLASE ESPECÍFICA:

Proyectos en la ventana, seleccione la clase que desea perfil. (Esta clase debe contener un ejecutable método.)

1. Seleccione Profile > Profile Other > Profile "*filename*" en el menú principal.
2. Seleccione profiling task en el cuadro de diálogo Perfiles de Tarea.
3. Especifique las opciones para la tarea.
4. Haga clic en Ejecutar.

Cuando haga clic en Ejecutar, sólo una selección de clase es lanzado.

Para el perfil de un proyecto o un archivo, el proyecto script debe modificarse para importar un script de perfiles. Si el script no contiene la tarea de importación de perfiles, el IDE modifica el script y crea una copia de seguridad del script original. Puede volver las modificaciones y eliminar la importación de perfiles tarea eligiendo Profile > Advanced Commands > Unintegrate Profiler from *project-name*

3.4.9 MENÚ VERSIONING

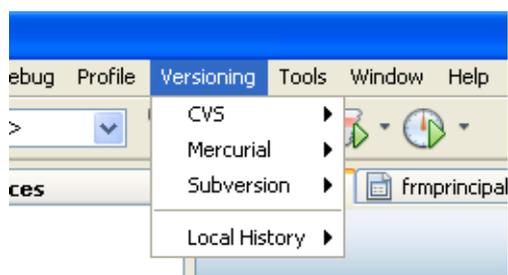


Figura 3. 35. Ventana MENÚ VERSIONING

En este menú tenemos las siguientes opciones:

- CVS
- Mercurial
- Subversion
- Local History

3.4.9.1 CVS: Versioning Options Window

El CVS proporciona una interfaz que le permite aplicar arbitraria metadatos a archivos o directorios por añadir, eliminar o modificar el formato de etiqueta para archivos versionados. También puede ajustar la configuración de CVS en el IDE, como la exclusión de los nuevos archivos de cometer acciones automáticamente, o limitar el número de caracteres por línea en cometer mensajes.

Para abrir la Versioning Opciones ventana, escogiendo Tools > Options del menú principal, entonces seleccione la opción Miscellaneous de la barra de navegación superior. Dentro de la categoría Miscellaneous, seleccione la etiqueta de Versioning localizada a lo largo de la parte superior del panel. Entonces, bajo los Sistemas de Versioning en el lado izquierdo del panel seleccione el CVS.

3.4.9.2 Mercurial Global Options

Este sistema ligero sirve para el manejo eficaz de proyectos distribuídos muy extensos, en trabajos Mercuriales con almacenamientos distribuídos normalmente se usan en muchos proyectos de la fuente abierta.

Los Plugin Mercuriales del IDE le permite manejar los cambios a los archivos controlados con los que trabaje. En el IDE, puede llamar los órdenes Mercuriales en directorios, proyectos, archivos y ventanas. El IDE también proporciona a un Espectador de Diff gráfico, permitiéndole comparar las revisiones del archivo, así como el diff del inline de apoyo directamente en el editor.

3.4.9.3 Subversion: Options Window

La ventana de Opciones de Subversión le permite configurar parámetros Subversión-específicos que efectúan la conducta de Subversión en el IDE.

Usted puede acceder este tablero escogiendo Tools > Options en el menú principal, escoja la categoría Miscellaneous, seleccione la etiqueta Subversión que se encuentra en el tablero izquierdo de la ventana.

3.4.9.4 Local History

El IDE proporciona que los versioning incorporados apoyan para los proyectos locales y archivos. Similar a los sistemas del versioning convencionales, el IDE mantiene una historia interior de recientes cambios hecha a las fuentes en su directorio activo local. Siempre que usted ahorre un archivo, esto es registrado como un 'comprometa' de una nueva versión del archivo en la historia local. Usted puede inspeccionar todas las revisiones en la historia local y Diff ellos contra los archivos actuales.

3.4.10 MENÚ TOOLS

En este menú Tools usted selecciona cualquiera de las escenas desplegadas en la ventana. Dependiendo de la categoría, que usted necesite.

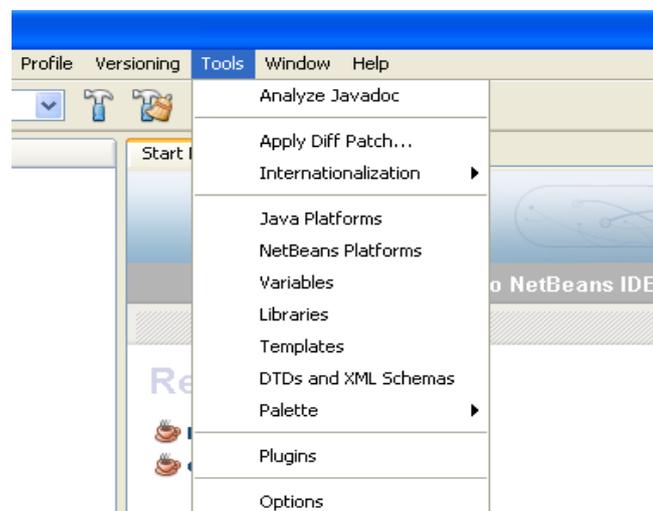


Figura 3. 36. Ventana MENÚ TOOLS

3.4.10.1 Comando Analyze Javadoc

Para cada uno de sus proyectos, usted puede producir un juego de Javadoc páginas de HTML que describen las clases del proyecto, clases internas, interfaz, constructores, métodos, y campos. El Javadoc se construye de la estructura de su código y el Javadoc comenta empotrado en su código. Usted también puede configurar cómo el IDE genera la documentación de Javadoc para cada uno de sus proyectos.

3.4.10.2 Comando Apply Diff Patch

Los archivos del parche habilitan a diseñadores del software que no están compartiendo un almacén común para distribuir e integrar cambios que se han hecho al código. El IDE le permite crear y aplicar parches que ponen al día que copia de archivos de la fuente para que usted no tenga que incorporar los cambios por mano.

3.4.10.3 Comando Internationalization

La internacionalización permite adaptar las aplicaciones a los varios idiomas y regiones sin requerir cambios de la ingeniería o recopilación. Los programas internacionalizados habilitan los elementos textuales, como los mensajes de estado y el componente de GUI etiqueta, para ser guardado fuera del código de la fuente y recuperó dinámicamente en lugar del hardcoded en el programa.

3.4.10.4 Comando NetBeans Platform

La Plataforma de NetBeans proporciona los requisitos comunes de una aplicación como los menús, dirección del documento, y escenas directamente de la caja. Construyendo una aplicación "Encima de NetBeans", en lugar de escribir las aplicaciones desde el principio, usted proporciona sólo las partes de su aplicación que la Plataforma de NetBeans ya no tiene. Al final del ciclo de desarrollo, usted ata su aplicación con la Plataforma de NetBeans, mientras ahorra tiempo y energía y produciendo una sólida aplicación fiable.

3.4.10.5 Comando Variables

La ventana de Variables Local muestra una lista de las variables locales en una llamada. Por defecto, la ventana de las Variables Locales abre automáticamente siempre que usted empiece una sesión de la depuración. Usted puede abrir la ventana de las Variables Locales escogiendo Window > Debugging > Local Variables.

Si usted abre la ventana de las Variables Locales cuando una sesión de la depuración está corriendo, cierra automáticamente cuando acaba la sesión de depuración. Si usted abre la ventana cuando ninguna sesión de la depuración está corriendo, se queda abierto hasta que usted lo cierre.

La información dada para cada variable incluye el nombre, tipo, y valor.

3.4.10.6 Comando Templates (Plantillas)

Para cada tipo de aplicación de Java, el IDE proporciona dos tipos de plantillas del proyecto:

3.4.10.6.1 Las plantillas normales

Plantillas en que el IDE controla toda la fuente y escenas del classpath, recopilación, funcionamiento, y depuración.

3.4.10.6.2 Las plantillas de Formulario libre

Plantillas en que su propia escritura, controla todas las escenas del classpath, recopilación, funcionamiento, y depuración.

3.4.10.7 Comando Palette (Paleta)

La ventana de la Paleta despliega una lista que contiene todos los componentes disponibles para agregar a los varios editores del IDE. Para los formularios de GUI, los artículos disponible en la Paleta los recipientes y otros componentes visuales, como los botones, etiquetas, menús, y tableros, incluyen.

Para abrir la ventana de la Paleta escogemos Window > Palette del menú principal.

3.4.10.8 Comando Plugins (otra vez)

El gerente de Plugins del IDE le permite poner al día y manejar el plugins de su IDE dinámicamente. Usted puede minimizar los startup cronometran y excepto la memoria dejando fuera de funcionamiento el plugins que usted no necesita. No se anulan los plugins desactivados de su directorio de la instalación, pero simplemente se ignora por el IDE. Usted puede re-activarlos cuando quiera a.

Usted también puede escoger borrar el plugins del IDE. Los plugins borrados están alejados de su directorio de la instalación. Para usar un plugins borrado, usted necesita instalarlo de nuevo.

Usted usa al gerente de Plugins para conectar a los centros de actualización para verificar si hay nuevos plugins o las nuevas versiones de plugins ya instalado disponible. Si nuevo o puso al día los plugins están disponibles, usted puede seleccionar, puede transmitir, y puede instalarlos usando al gerente de Plugins.

Un plugin generalmente consiste en un grupo de módulos dependientes. Cuando usted pone al día un plugin, la actualización normalmente consiste en actualización uno o más de los módulos que constituyen el plugin. Normalmente dejando fuera de funcionamiento un plugin consiste en dejar fuera de funcionamiento los módulos individuales que constituyen el plugin.

3.4.10.9 Comando Options (Opciones)

La ventana de las Opciones es donde el usuario define escenas, como la situación del navegador de tejido usada por el IDE. La categoría puede desplegar los tableros:



Figura 3. 37. Cuadro de Opciones de Comando

- General
- Editor
- Fonts & Colors
- Keymap
- Miscellaneous.

3.4.11 MENÚ WINDOW

Este es un menú de acceso.

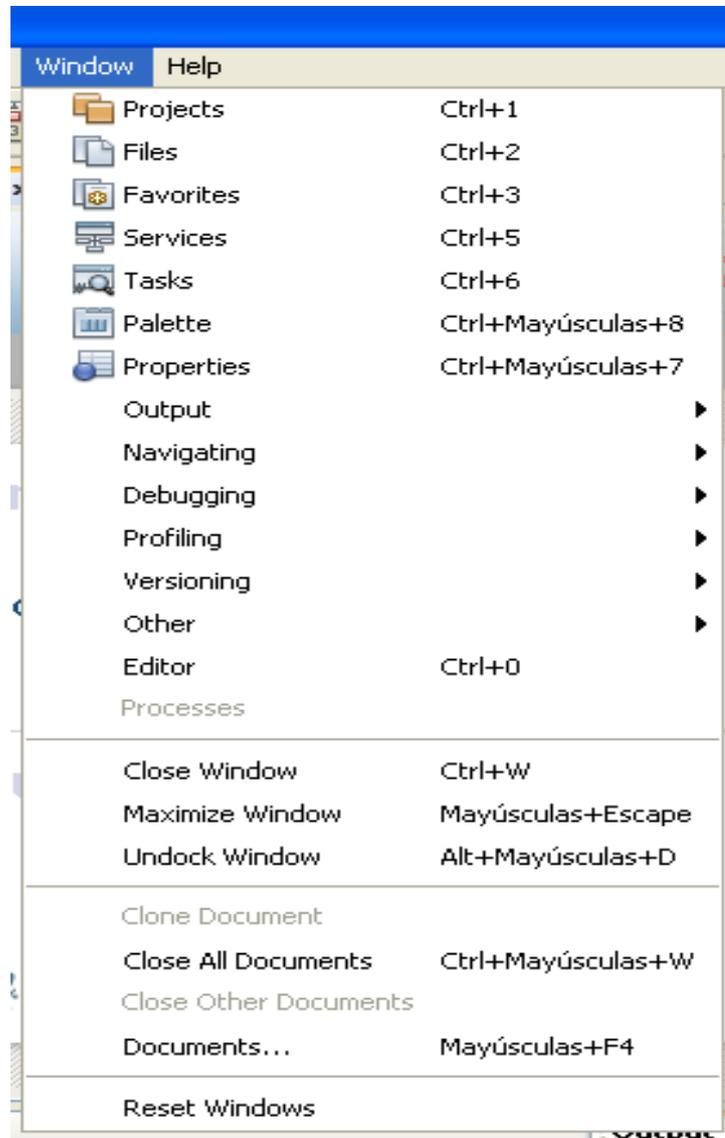


Figura 3. 38. Ventana MENÚ WINDOW

Keys	Command	Action
Ctrl-0	Source Editor	Los interruptores al Editor de la Fuente.
Ctrl-1/Ctrl-Shift-1	Projects	Abre la ventana de los Proyectos.
Ctrl-2/Ctrl-Shift-2	Files	Abre la ventana de los Archivos.
Ctrl-3/Ctrl-Shift-3	Favorites	Abre la ventana de los Favoritos.
Ctrl-4	Output Window	Abre la ventana del Rendimiento.
Ctrl-5	Services	Abre la ventana de Servicios.
Ctrl-Shift-5	HTTP Monitor	Abre al Amonestador de HTTP.
Ctrl-6	Task List	Abre la ventana de Lista de Tarea.
Ctrl-7	Navigator	Abre al Navegante.
Alt-Shift-1	Debugging > Local Variables	Abre la ventana de depurador de Variables Local.
Alt-Shift-2	Debugging > Watches	Abre la ventana de depurador de Relojes.
Alt-Shift-3	Debugging > Call Stack	Abre la Llamada Pila depurador ventana.
Alt-Shift-4	Debugging > Classes	Abre la ventana de depurador de Clases.
Alt-Shift-5	Debugging > Breakpoints	Abre la ventana de depurador de Puntos de ruptura.
Alt-Shift-6	Debugging >	Abre la ventana de depurador de

	Sessions	Sesiones.
Alt-Shift-7	Debugging > Threads	Abre la ventana de depurador de Hilos.
Alt-Shift-8	Debugging > Sources	Abre la ventana de las Fuentes.
Ctrl-W	Close	Los cierras la etiqueta actual en la ventana actual. Si la ventana no tiene ninguna etiqueta, la ventana entera está cerrada.
Shift-Escape	Maximize Window	Aumenta al máximo el Editor de la Fuente o la ventana presente.
Alt-Shift-D	Undock Window	Destaca la ventana del IDE.
Ctrl-Shift-w	Close All Documents	Cierra todos los documentos abiertos en el Editor de la Fuente.
Shift-F4	Documents	Abre los Documentos y usted puede ahorrar y grupos íntimos de documentos abiertos.
Ctrl-Tab (Ctrl-')	Switch to Recent Window	La caja del diálogo despliega todas las ventanas abiertas y cada uno de los documentos abiertos en el Editor de la Fuente.

3.5 PRESENTACIONES DE EJEMPLOS EN LA PLATAFORMA.

3.5.1 EJEMPLO1: INCREMENTO Y DECREMENTO

Ejemplo de Operadores Aritméticos de Incremento y decremento.

```
public class Ejemplo1{

    public static void main(String[] args) {

        int i = 1;

        prt("i : " + i);

        prt("++i : " + ++i); // Pre-incremento, primero
        //incrementa y luego imprime por consola

        prt("i++ : " + i++); // Post-incremento, primero imprime
        //"2" por consola y luego incrementa i.

        prt("i : " + i);//i por lo tanto vale 3

        prt("--i : " + --i); // Pre-decremento, primero
        //decrementa i y luego lo imprime por consola

        prt("i-- : " + i--); // Post-decremento, primero imprime
        prt("i : " + i);//Ahora i vale 1

    }

    //esto nos ahorrara teclear. Invocando a prt podremos
    //imprimir por consola la cadena de caracteres que le pasemos

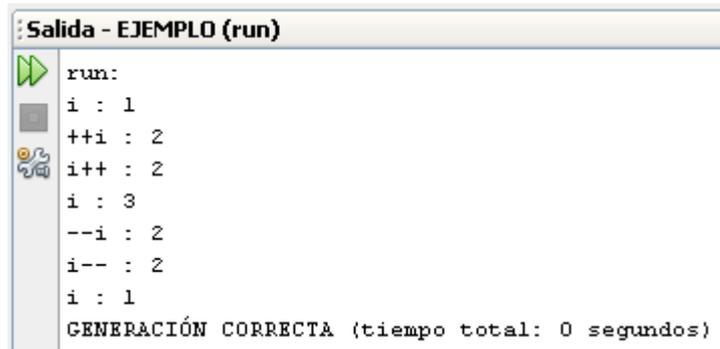
    static void prt(String s) {

        System.out.println(s);

    }

}
```

Su salida es la siguiente:



```
run:
i : 1
++i : 2
i++ : 2
i : 3
--i : 2
i-- : 2
i : 1
GENERACIÓN CORRECTA (tiempo total: 0 segundos)
```

3.5.2 EJEMPLO2: OPERADORES LÓGICOS

Con el siguiente ejemplo se muestra la función de operadores lógicos:

```
import java.util.*;

public class Ejemplo2 {

    public static void main(String[] args) {

        //Creamos un objeto de tipo Random, almacenado un puntero a
        //el en la variable rand.

        Random rand = new Random();

        //el método nextInt() del objeto Random creado (se invoca como
        //rand.nextInt()) genera un número aleatorio entero.
        //El módulo 100 de un entero aleatorio será un entero
        //aleatorio entre 0 y 100.

        int i = rand.nextInt() % 100;

        int j = rand.nextInt() % 100;

        //Imprime i y j por consola

        prt("i = " + i);

        prt("j = " + j);
    }
}
```

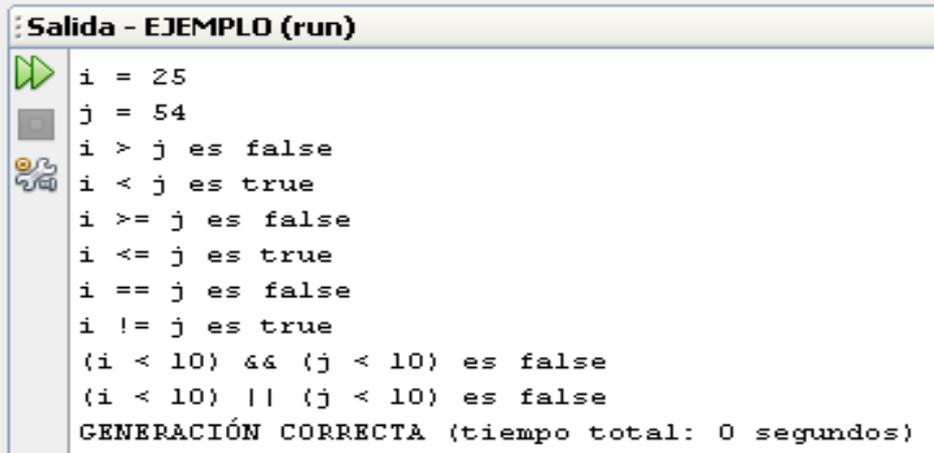
//Imprime diversas operaciones binarias sobre i y j, junto

//con su resultado.

```
prt("i > j es " + (i > j));  
prt("i < j es " + (i < j));  
prt("i >= j es " + (i >= j));  
prt("i <= j es " + (i <= j));  
prt("i == j es " + (i == j));  
prt("i != j es " + (i != j));  
prt("(i < 10) && (j < 10) es "  
+ ((i < 10) && (j < 10)) );  
prt("(i < 10) || (j < 10) es "  
+ ((i < 10) || (j < 10)) );  
}
```

```
static void prt(String s) {  
    System.out.println(s);  
}  
}
```

Una posible salida de este programa es:



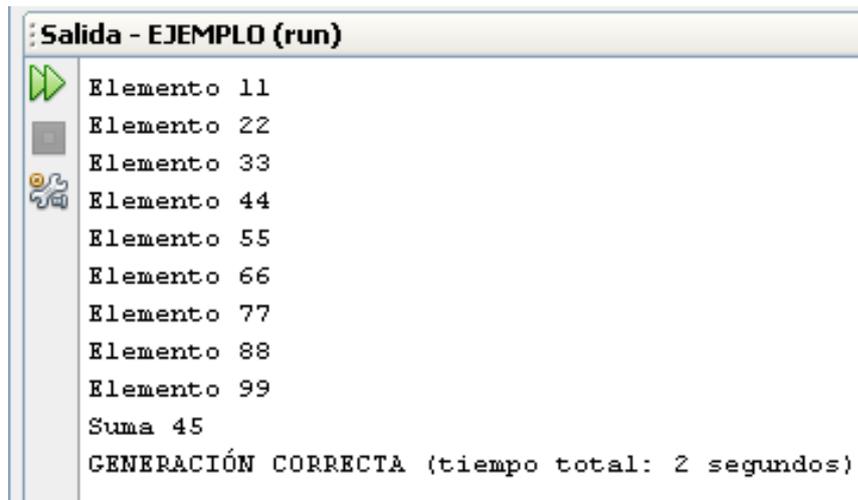
```
Salida - EJEMPLO (run)  
i = 25  
j = 54  
i > j es false  
i < j es true  
i >= j es false  
i <= j es true  
i == j es false  
i != j es true  
(i < 10) && (j < 10) es false  
(i < 10) || (j < 10) es false  
GENERACIÓN CORRECTA (tiempo total: 0 segundos)
```

3.5.3 EJEMPLO3: “ARRAY”

En este ejemplo hemos definido un array llamado alumnos, en el que podremos almacenar 10 datos tipo entero. El primer elemento de un array se sitúa en la posición 0, exactamente igual que en C. Si queremos realizar un bucle que recorra los elementos de este array escribimos el siguiente código:

```
public class Ejemplo3 {  
    public static void main(String[] args) {  
        int[] = new int[10];  
        for(int i= 0; i< 10; i++){  
            alumnos[i] = i;  
            System.out.println("Elemento " + i + alumnos[i]);  
        }  
        int sum = 0;  
        for(int i= 0; i< 10; i++){  
            sum = sum + alumnos[i];  
        }  
        System.out.println("Suma " + sum);  
    }  
}
```

La salida del programa es:



```
Salida - EJEMPLO (run)
Elemento 11
Elemento 22
Elemento 33
Elemento 44
Elemento 55
Elemento 66
Elemento 77
Elemento 88
Elemento 99
Suma 45
GENERACIÓN CORRECTA (tiempo total: 2 segundos)
```

3.5.4 EJEMPLO4: SENTENCIA “IF”

Ejemplo de la sentencia if , si condición toma el valor true se ejecuta Grupo de sentencias, si condicion2 toma el valor true se ejecuta Grupo2 de sentencias... y así sucesivamente hasta acabarse todas las condiciones. Si no se cumple ninguna se ejecuta Grupo_n de sentencias. Este último else es opcional. En ambos casos se continúa ejecutando el resto del código. En el siguiente ejemplo lo entenderán mejor.

```
public class Ejemplo4 {
    // Método que podremos invocar como test(int a, int b) y que
    // devolverá -1 si a < b, +1 si a > b y 0 si a == b.
    static int test(int val, int val2) {
        int result = 0;
        if(val > val2)
            result = +1;
        else if(val < val2)
            result = -1;
```

```

        else

            result = 0;

            return result;

        }

    public static void main(String[] args) {

        //Imprimimos por consola el resultado de realizar unos

        //cuantos test.

        System.out.println(test(10, 5));

        System.out.println(test(5, 10));

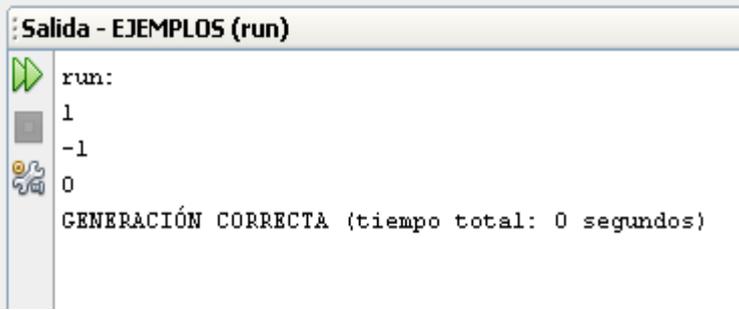
        System.out.println(test(5, 5));

    }

}

```

La salida del programa es la siguiente:



```

run:
1
-1
0
GENERACIÓN CORRECTA (tiempo total: 0 segundos)

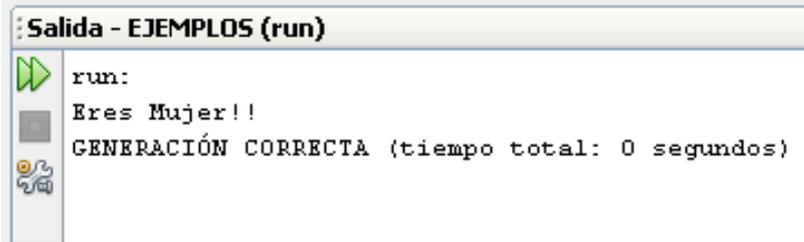
```

3.5.5 EJEMPLO 5: SENTENCIA “SWITCH”

En este ejemplo utilizaremos la sentencia switch, vamos a usar char, con el cual determinaremos si es H hombre o M mujer ingresando solo una letra.

```
public class Ejemplo5{  
    public static void main(String[] args){  
        char sexo='F';  
        switch(sexo){  
            case 'F':  
                System.out.println("Eres Mujer!!");  
                break;  
            case 'M':  
                System.out.println("Eres Hombre!!");  
                break;  
            case 'I':  
                System.out.println("Como que Indiferente, raro XD");  
                break;  
            default:  
                System.out.println("Elija un sexo Valido");  
        }  
    }  
}
```

La salida del programa es:



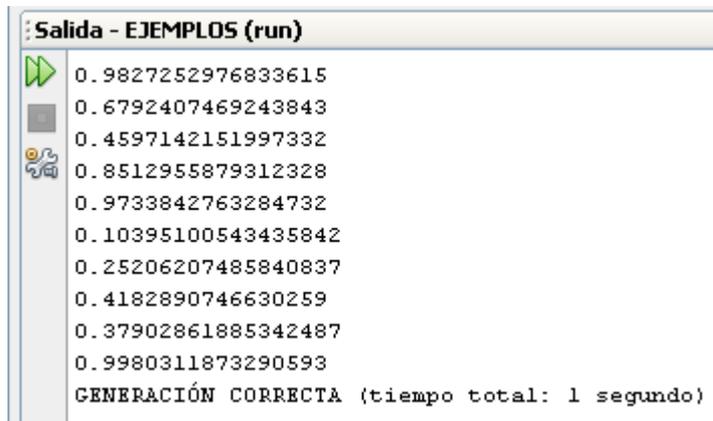
```
Salida - EJEMPLOS (run)  
run:  
Eres Mujer!!  
GENERACIÓN CORRECTA (tiempo total: 0 segundos)
```

3.5.6 EJEMPLO6: SENTENCIA “WHILE”

Con el siguiente ejemplo analizamos la función del bucle while donde se comprueba si se verifica su condición, si se verifica se continua ejecutando el código del bucle hasta que esta deje de verificarse.

```
class Ejemplo6 {  
    public static void main(String[] args) {  
        double r = 0;  
  
        //Mientras que r < 0.99 sigue ejecutando el cuerpo del bucle.  
  
        //La d significa double. No es necesaria  
  
        while(r < 0.99d) { //Genera un nuevo r aleatorio entre 0 y 1.  
            r = Math.random(); //Lo imprime por consola.  
            System.out.println(r);  
        }  
    }  
}
```

La salida del programa es:



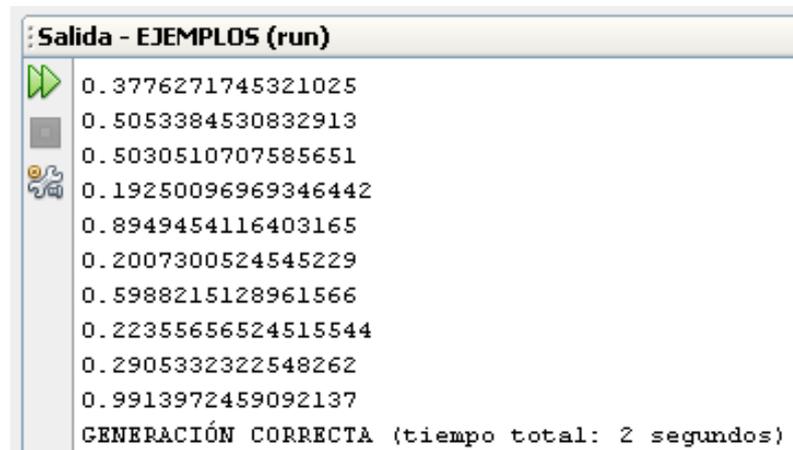
```
: Salida - EJEMPLOS (run)  
0.9827252976833615  
0.6792407469243843  
0.4597142151997332  
0.8512955879312328  
0.9733842763284732  
0.10395100543435842  
0.25206207485840837  
0.4182890746630259  
0.37902861885342487  
0.9980311873290593  
GENERACIÓN CORRECTA (tiempo total: 1 segundo)
```

3.5.7 EJEMPLO7 : SENTENCIA “DO WHILE”

El ejemplo anterior implementado mediante un bucle do while independientemente del valor de r ejecutará al menos una vez el código de su cuerpo.

```
public class Ejemplo7 {  
    public static void main(String[] args) {  
        double r;  
  
        //Idéntico al ejemplo anterior, solo que ahora la condición  
        //está al final del bucle.  
  
        do {  
            r = Math.random();  
            System.out.println(r);  
        } while(r < 0.99d);  
    }  
}
```

La salida del programa es:



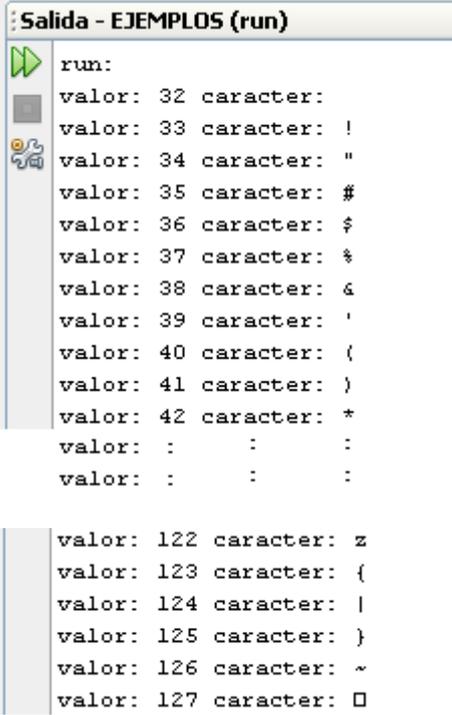
```
Salida - EJEMPLOS (run)  
0.3776271745321025  
0.5053384530832913  
0.5030510707585651  
0.19250096969346442  
0.8949454116403165  
0.2007300524545229  
0.5988215128961566  
0.22355656524515544  
0.2905332322548262  
0.9913972459092137  
GENERACIÓN CORRECTA (tiempo total: 2 segundos)
```

3.5.8 EJEMPLO8 : SENTENCIA “FOR”

Programa que imprime los caracteres correspondientes a los números enteros comprendidos entre 32 y 128 utilizando el bucle for.

```
public class Ejemplo8{  
    public static void main(String[] args) {  
        for( char c = 32; c < 128; c++)  
            //Imprime los caracteres correspondientes a los números  
            //enteros comprendidos entre 32 y 128. (int)c es el entero  
            //correspondiente al carácter c.  
            System.out.println("valor: " + (int)c +  
                " caracter: " + c);  
    }  
}
```

La salida del programa es:



```
run:  
valor: 32 caracter:   
valor: 33 caracter: !  
valor: 34 caracter: "  
valor: 35 caracter: #  
valor: 36 caracter: $  
valor: 37 caracter: %  
valor: 38 caracter: &  
valor: 39 caracter: '  
valor: 40 caracter: (  
valor: 41 caracter: )  
valor: 42 caracter: *  
valor: : : :  
valor: : : :  
  
valor: 122 caracter: z  
valor: 123 caracter: {  
valor: 124 caracter: |  
valor: 125 caracter: }  
valor: 126 caracter: ~  
valor: 127 caracter: □
```

3.6 DISEÑO DE APLICACIÓN WEB

3.6.1 APLICACIONES WEB EN NETBEANS

Este ejemplo muestra cómo crear una aplicación web simple, su despliegue en el servidor y su presentación en un navegador. La aplicación utiliza una página JavaServer Pages™ (JSP) para que ingrese su nombre. Luego utiliza un componente JavaBeans™ para que el nombre persista durante toda la sesión HTTP, y devuelve el nombre en la segunda página JSP.

Para empezar, vamos a crear un proyecto web en nuestro NetBeans:

3.6.1.1 Configurando un Proyecto de Aplicación Web

Escoja File > New Project (Ctrl-Shift-N) del menú principal. En Categories, seleccione Web. Bajo Projects, seleccione Web Application entonces haga click en Next.

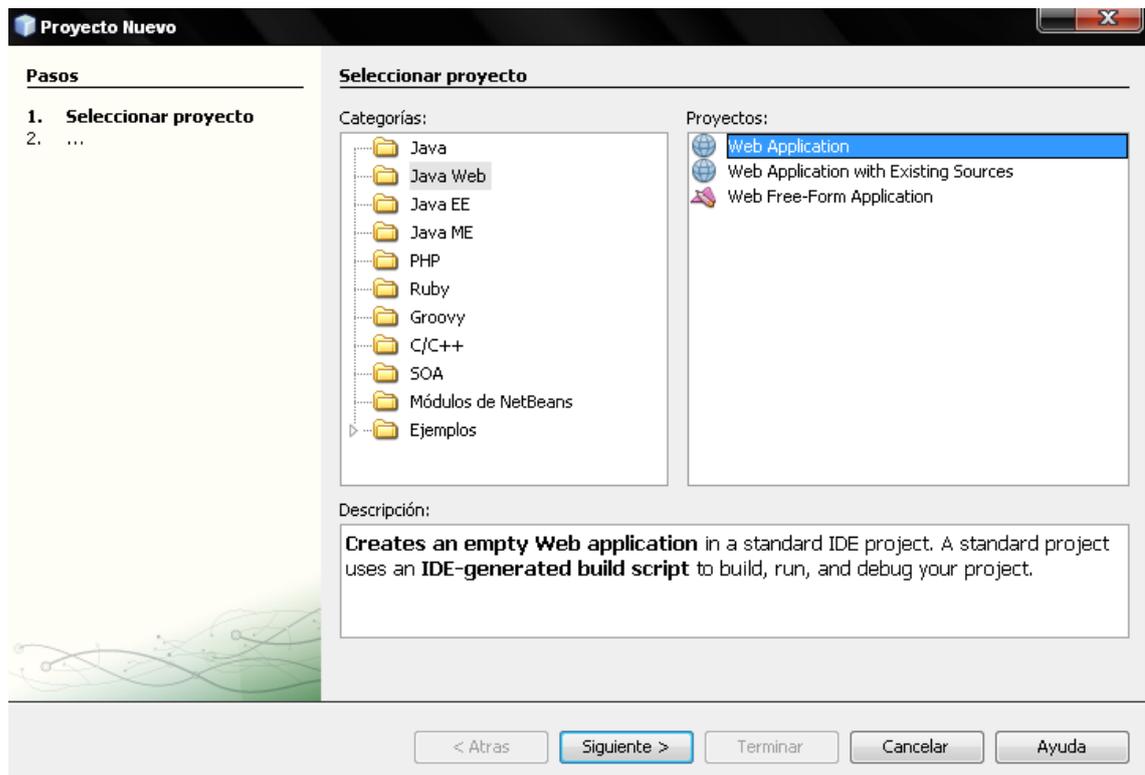


Figura 3.5. 1. Ventana Proyecto Nuevo Web Application

Introduzca Hola_Web en el campo Project Name. Note que el Context Path (e.j., en el servidor) se convierte en /Hola_Web.

Especifique en Project Location, cualquier directorio en su computador.

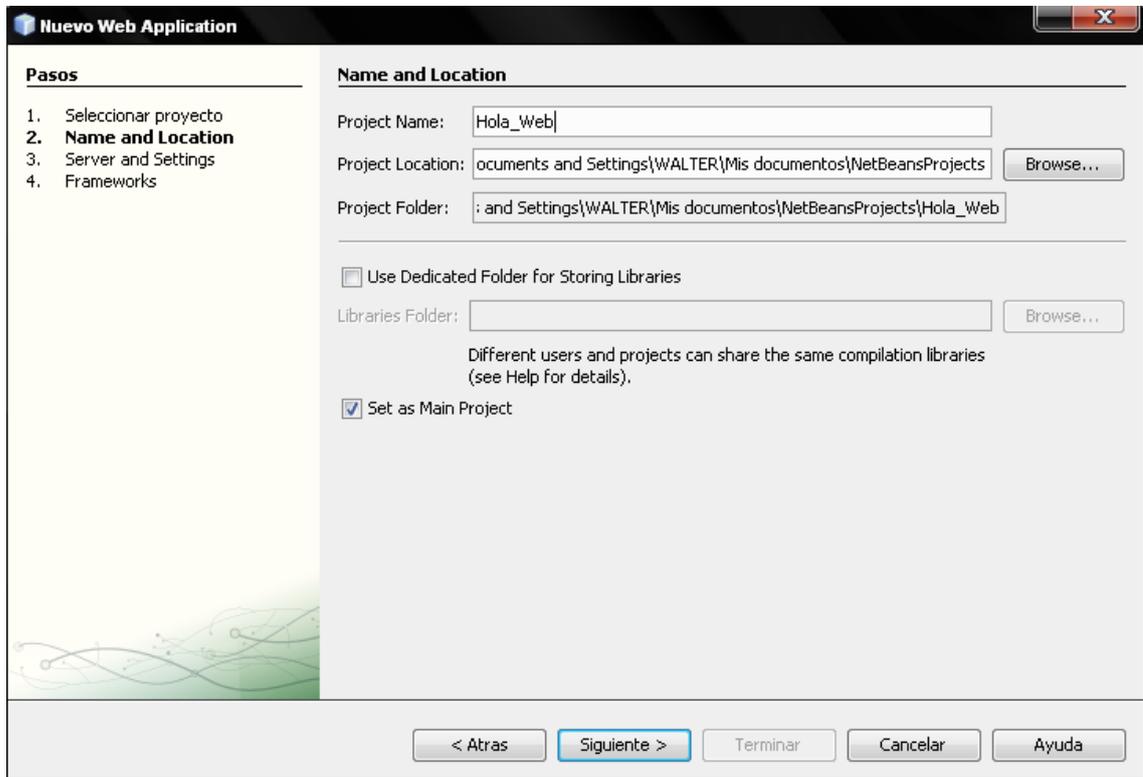


Figura 3.5. 2. Ventana Nueva Web Application

Nota: La creación de proyectos en NetBeans IDE 6.5 incluye nuevas opciones que se pueden dejar por defecto. Por ejemplo, puede dejar sin marcar la casilla Use Dedicated Folder for Storing Libraries.

Si está utilizando NetBeans IDE 6.5, haga click en Next.

Seleccione el servidor en el cual va a desplegar su aplicación. Solamente se listarán servidores que estén registrados en el IDE. Haga click en siguiente:

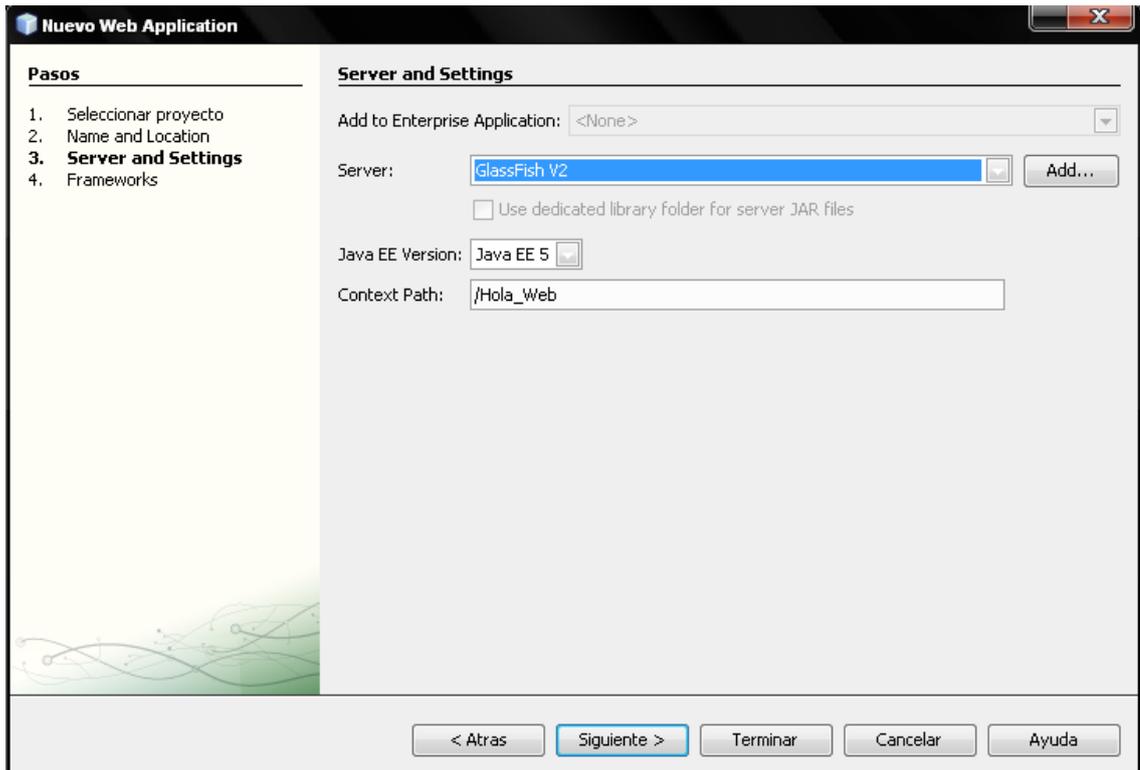


Figura 3.5. 3. Ventana Nuevo Web Application Server and Setting

En el panel de Frameworks, haga click en Finish para crear el proyecto.

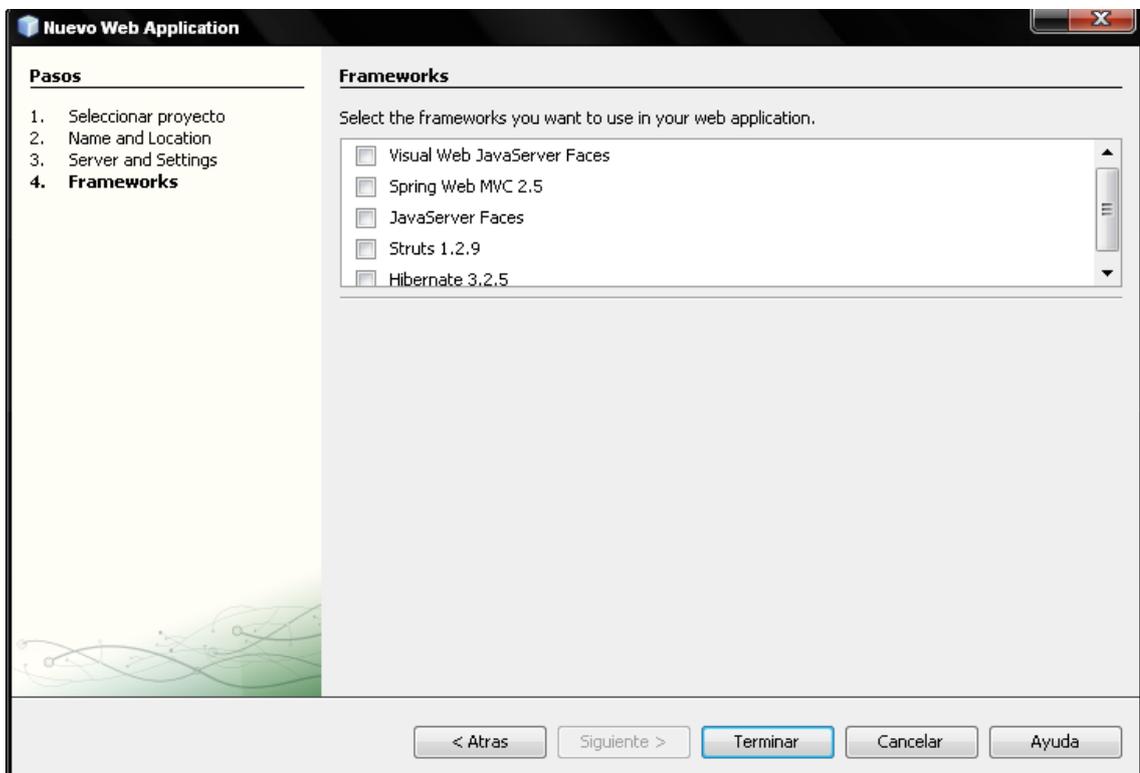
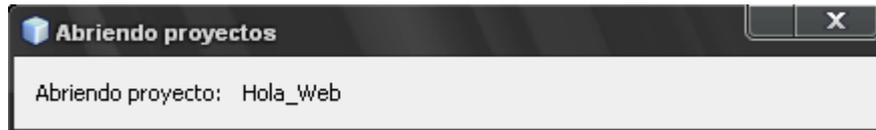


Figura 3.5. 4. Ventana Nuevo Web Application Frameworks



La carpeta del proyecto contiene todas las fuentes y metadata del proyecto, como el script de compilación Ant. El proyecto Hola_Web se abre en el IDE. La página de bienvenida, index.jsp, se abre en el Source Editor de la ventana principal. Podrá ver la estructura de archivos del proyecto en la ventana Files (CTRL-2), y su estructura lógica en la ventana Projects (CTRL-1):

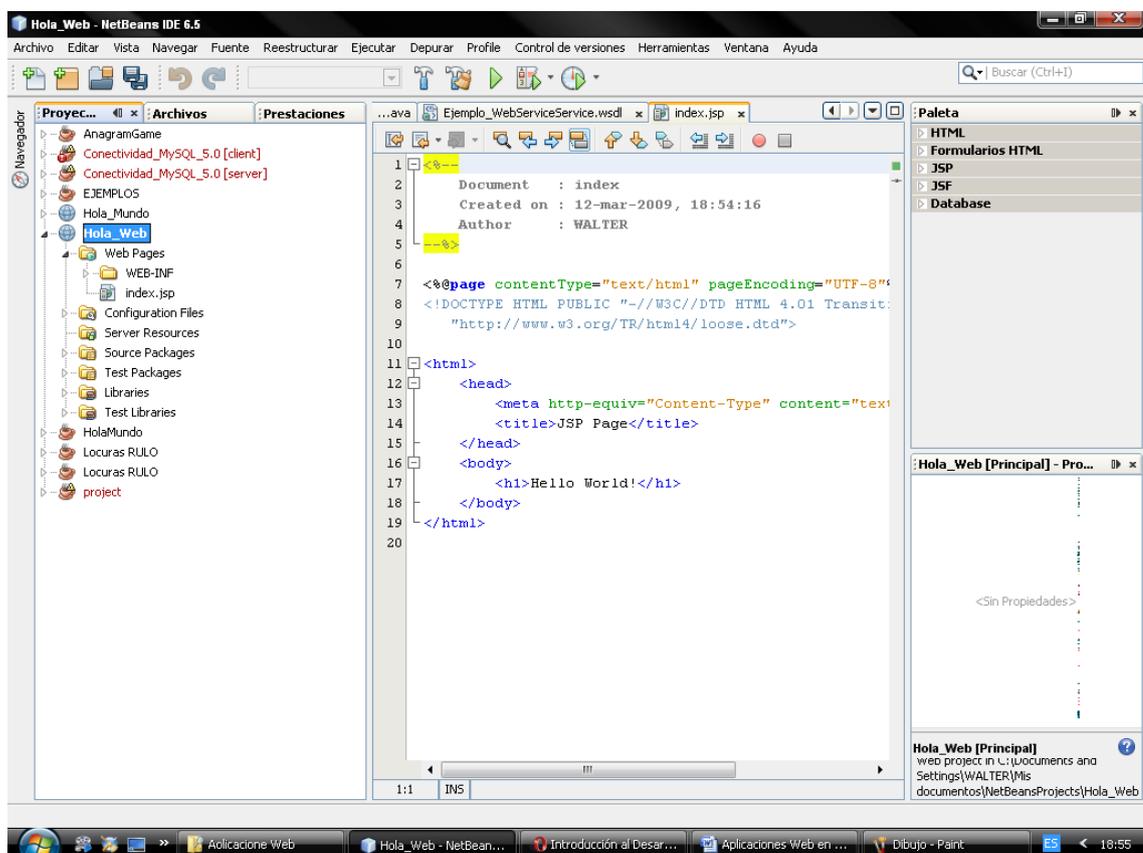


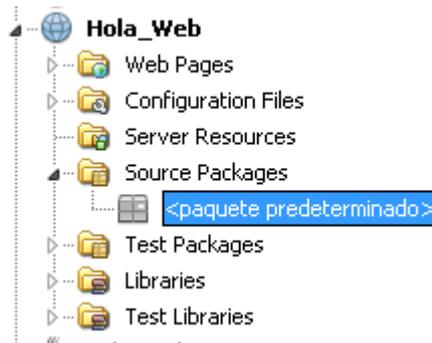
Figura 3.5. 5. Ventana Proyecto Hola Web

3.6.2 CREANDO Y EDITANDO LOS ARCHIVOS FUENTE DE LA APLICACIÓN WEB

La creación y edición de archivos fuente es la función más importante que proporciona el IDE. Claro está que es la actividad a la que le dedica más tiempo. El IDE proporciona una amplia variedad de herramientas que pueden complementar el estilo personal de cualquier desarrollador, aun si prefiere escribir todo a mano o si espera que el IDE genere el código por usted.

Creando un Paquete Java y un Archivo Fuente Java

- En la ventana Projects, expanda el nodo Source Packages. Note que el nodo Source Packages solo contiene un nodo por de facto vacío.



- Haga click derecho en el nodo Source Package y seleccione New > Java Class. Ingrese NameHandler en el campo Class Name y escriba org.mypackage.hello en el campo Package. Haga click en Finish. Note que el nuevo archivo NameHandler.java se abre en Source Editor.

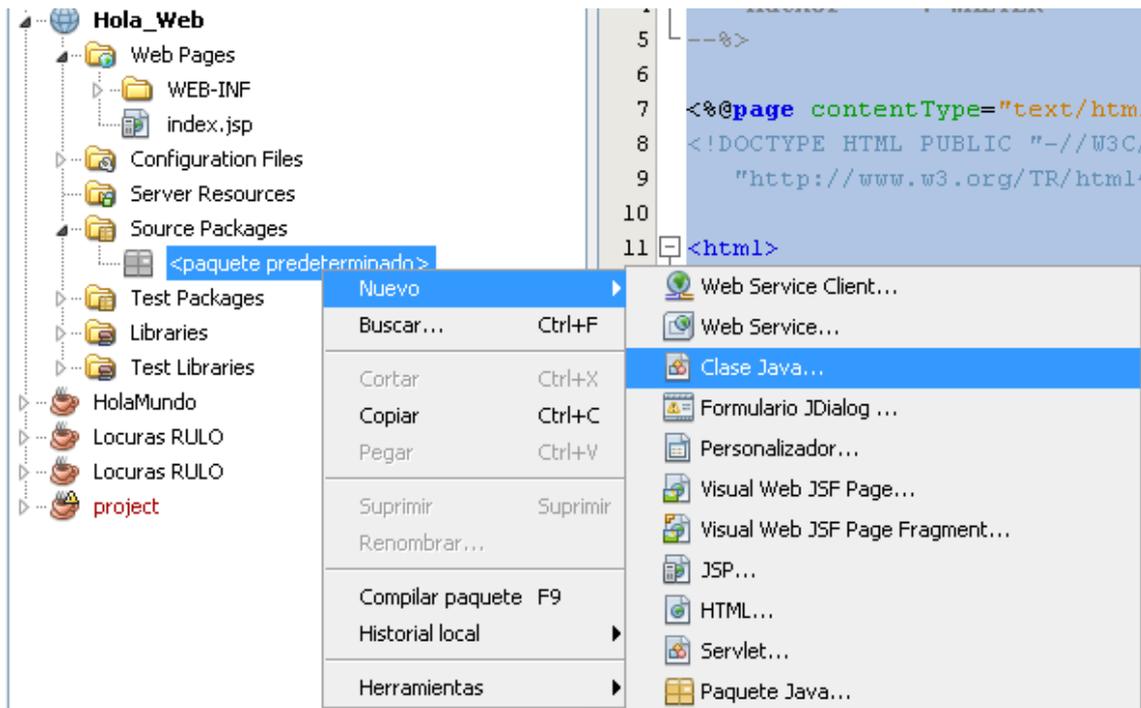


Figura 3.5. 6. Ventana Nuevo Class Java

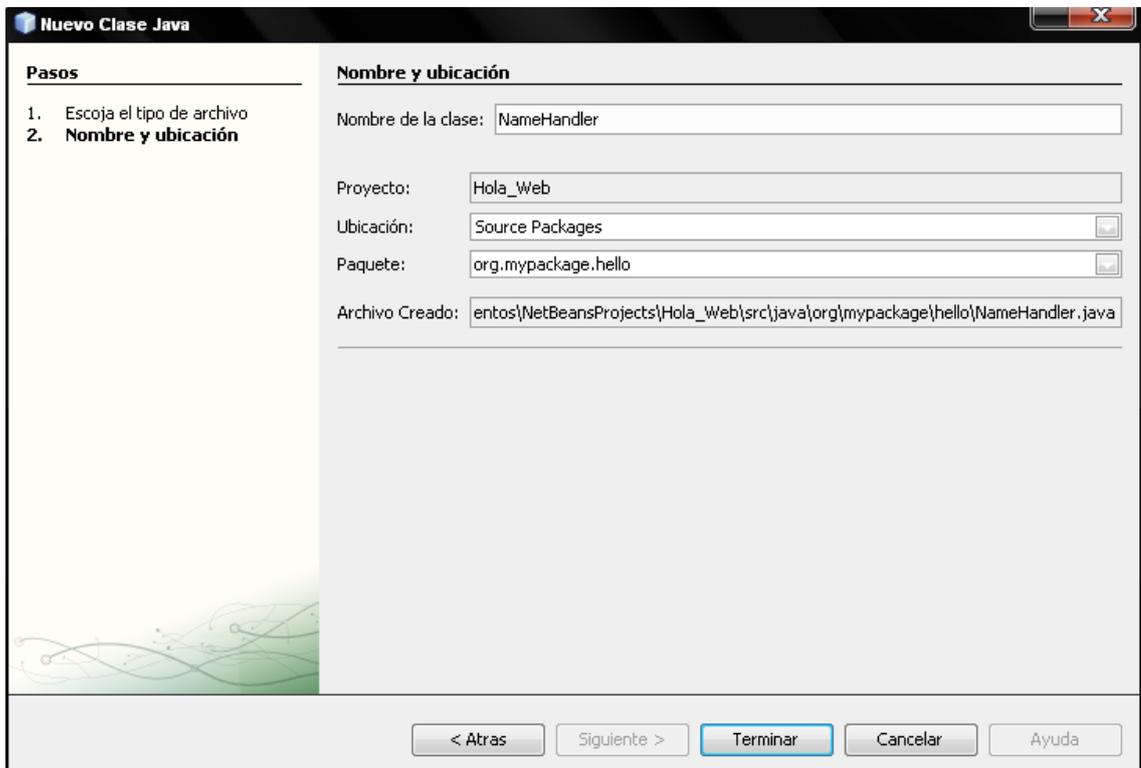


Figura 3.5. 7. Ventana Nuevo Class Java Nombre y Ubicación

- En Source Editor, declare una variable String escribiendo la siguiente línea directamente debajo de la declaración de la clase:

```
String name;
```

- Añada el siguiente constructor a la clase:

```
public NameHandler()
```

- Añada la siguiente línea en el constructor NameHandler():

```
name = null;
```

3.6.3 GENERANDO MÉTODOS GETTER Y SETTER

Haga click derecho en el campo name en Source Editor y seleccione Refactor > Encapsulate. Se abre el dialogo Encapsulate Fields, listado el campo name. Note que Fields' Visibility está por de facto establecido como private, y Accessors' Visibility como public, indicando que el modificador de acceso para la declaración de variable de clase será especificada como privada, mientras que los métodos getter y setter se generaran con modificadores public y private respectivamente.

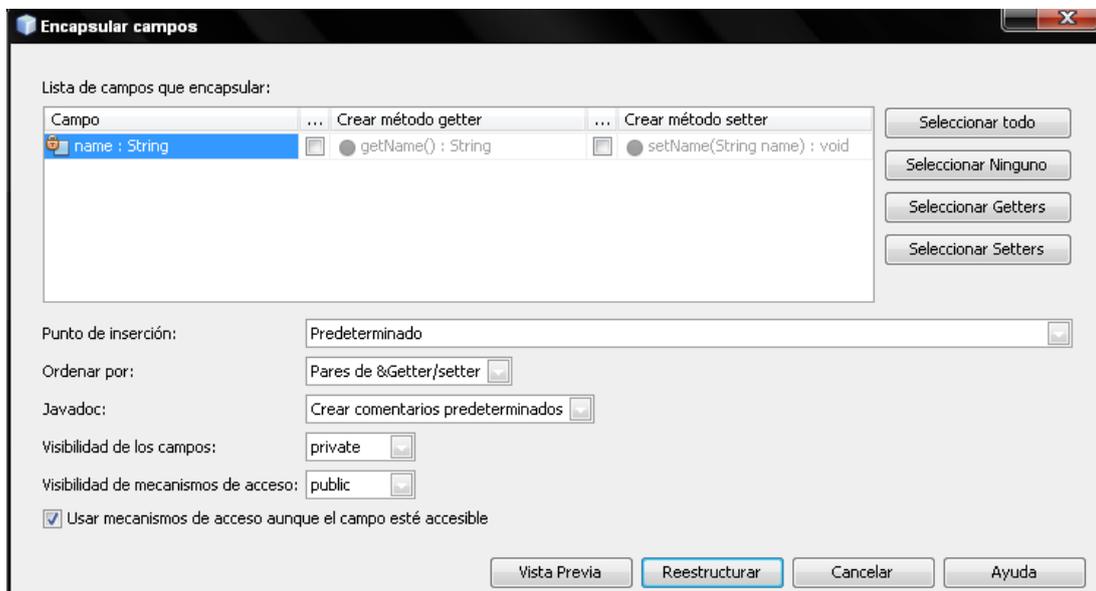


Figura 3.5. 8. Ventana Encapsular campos

Haga click en Refactor. Los métodos getter y setter son generados para el campo name. El modificador para la variable de clase está definido como private mientras que los métodos getter y setter son generados como modificadores públicos. La clase Java ahora se verá similar a lo siguiente:

```
package org.mypackage.hello;
```

```
1  /*
2   * To change this template, choose Tools | Templates
3   * and open the template in the editor.
4   */
5
6  package org.mypackage.hello;
7
8  /**
9   *
10  * @author WALTER
11  */
12  public class NameHandler {
13      private String name;
14
15      public NameHandler() {
16          name=null;
17      }
18
19      public String getName() {
20          return name;
21      }
22
23      public void setName(String name) {
24          this.name = name;
25      }
26  }
27
```

3.6.4 EDITANDO EL ARCHIVO JSP (JAVASERVER PAGES)

- Retorne al archivo index.jsp haciendo click en la lengüeta disponible sobre Source Editor.
- En Palette (Ctrl-Shift-8) localizada en la parte derecha de Editor Source, expanda HTML Forms y arrastre el ítem Form dentro de Source Editor, a un punto después de las etiquetas <h2>. La caja de dialogo de Insert Form se muestra así:

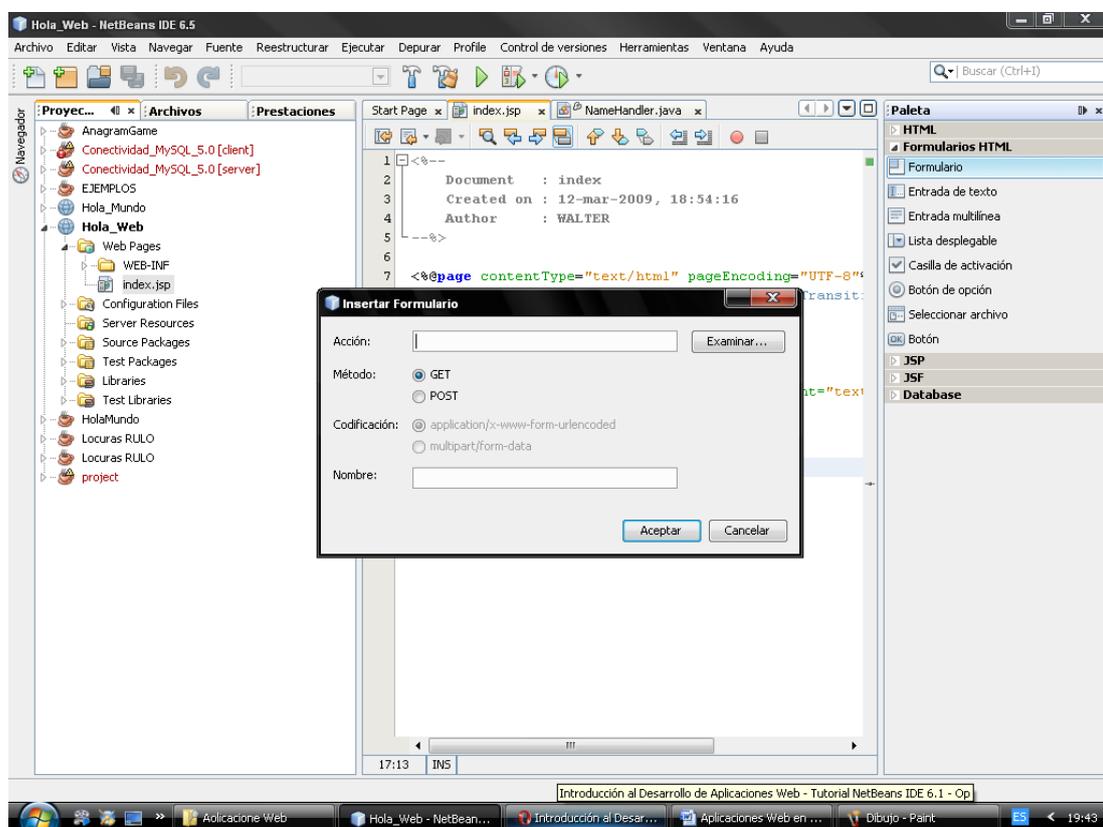


Figura 3.5. 9. Ventana Hola_Web Palette

- Especifique los siguientes valores:
 - **Action:** response.jsp
 - **Method:** GET
 - **Name:** Name Input Form

Haga click en OK. Un formulario HTML es añadido al archivo index.jsp.

- Arrastre un ítem Text Input a un punto antes de la etiqueta `</form>`, entonces especifique los siguientes valores:
 - **Name:** name
 - **Type:** text

Haga click en OK. Una etiqueta HTML `<input>` ha sido añadida dentro de las etiquetas `<form>`.

- Arrastre un ítem Button a un punto justo antes de la etiqueta `</form>`. Especifique los siguientes valores:
 - **Label:** OK
 - **Type:** submit

Haga clic en OK. Un botón HTML a sido añadido entre las etiquetas `<form>`.

Escriba Ingrese su nombre: justo antes de la etiqueta `<input>`, entonces cambie el texto Hola Mundo! que está entre las etiquetas `<h2>` a Entry Form.

Haga click derecho dentro de Source Editor y seleccione Format (Alt-Shift-F) para limpiar el formato de su código. Su archivo index.jsp se parecerá a lo siguiente:

```
1 <!--
2     Document    : index
3     Created on  : 12-mar-2009, 18:54:16
4     Author     : WALTER
5 -->
6
7 <%@page contentType="text/html" pageEncoding="UTF-8"%>
8 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
9     "http://www.w3.org/TR/html4/loose.dtd">
10
11 <html>
12     <head>
13         <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
14         <title>JSP Page</title>
15     </head>
16     <body>
17         <h1>Hello Mundo</h1>
18         <form name="Name Input Form" action="responde.jsp">
19
20             Ingrese su nombre:
21             <input type="text" name="name" value="" />
22             <input type="button" value="OK" />
23         </form>
24     </body>
25 </html>
26
```

3.6.5 CREANDO UN ARCHIVO JSP (JAVASERVER PAGES)

- En la ventana Projects, haga click derecho en el nodo del proyecto HelloWeb y seleccione New > JSP. El asistente New JSP File se abrirá. Dele el nombre response, y haga click en Finish. Note que el nodo del archivo response.jsp se muestra en la ventana Projects debajo de index.jsp, y el nuevo archivo se abre en Source Editor.

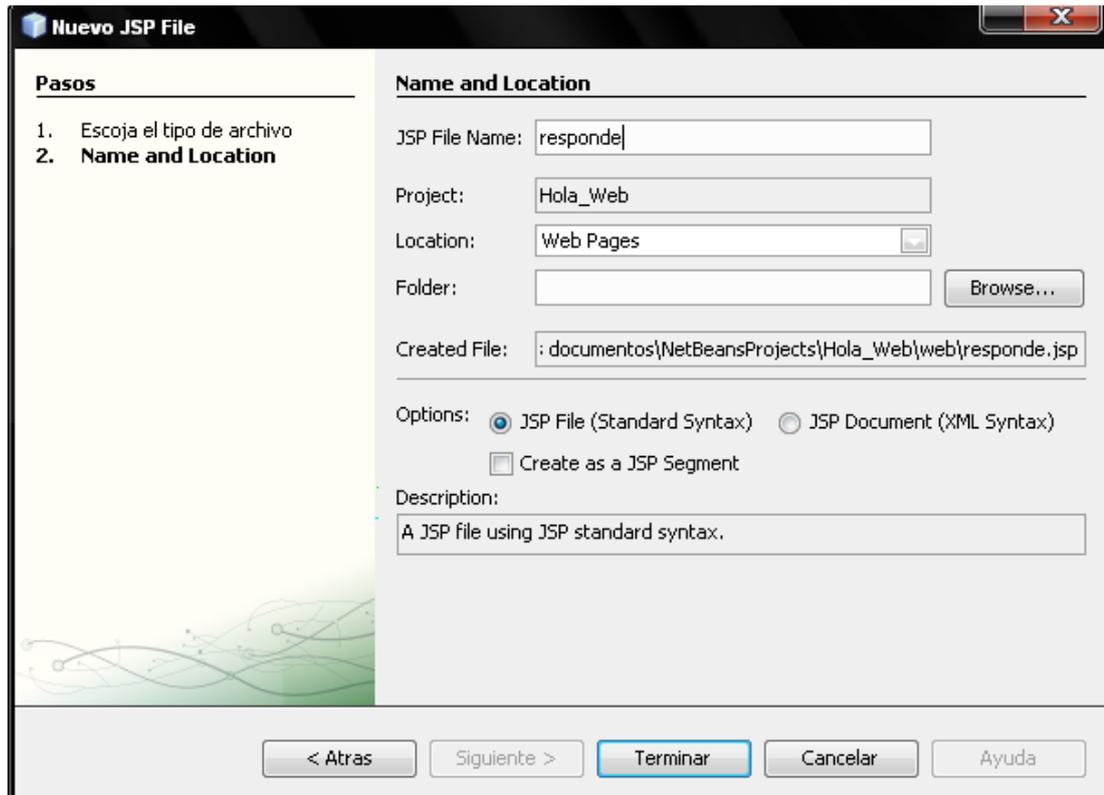


Figura 3.5. 10. Ventana Nuevo JSP File

- En Palette a la derecha de Source Editor, expanda JSP y arrastre el ítem Use Bean a un punto debajo de la etiqueta <body> en Source Editor. La caja de dialogo de Insert Use Bean se abre. Ingrese los siguientes valores:
 - **ID:** mybean
 - **Class:** org.mypackage.hello.NameHandler
 - **Scope:** session

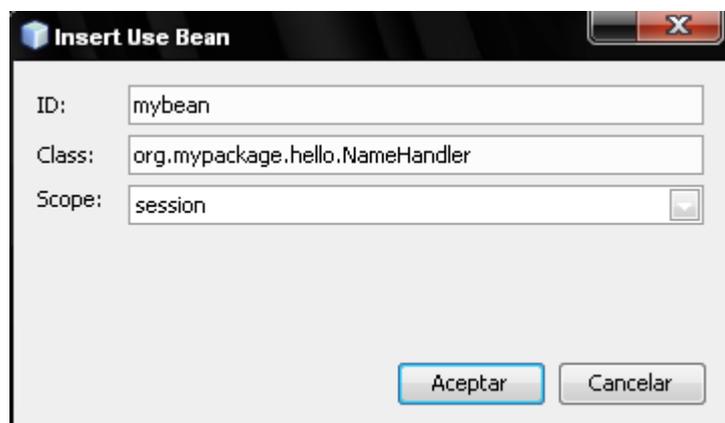


Figura 3.5. 11. Ventana Insert Use Bean

Haga click en OK. Note que las etiquetas `<jsp:useBean>` aparecen debajo de la etiqueta `<body>`.

- Arrastre el ítem Set Bean Property de Palette a un punto antes de la etiqueta `<h2>` y haga click en OK. En la etiqueta `<jsp:setProperty>`, borre el atributo vacío `value` y cámbielo por lo siguiente:

```
<jsp:setProperty name="mybean" property="name" />
```

Como se indica en la documentación de `<jsp:setProperty>`, puede indicar un valor de propiedad de varias maneras. En este caso, el ingreso del usuario que viene de `index.jsp` se convierte en un par nombre/valor que pasa al objeto `request object`. Cuando define una propiedad usando la etiqueta `<jsp:setProperty>`, podrá especificar el valor de acuerdo con el nombre de la propiedad contenida en el objeto `request`. Por tanto, al definir `property` a `name`, podrá recuperar el valor especificado por el ingreso del usuario.

- Cambie el texto entre las etiquetas `<h2>` para que se vea como los siguiente:

```
<h2>Hola, !</h2>
```

- Arrastre el ítem Get Bean Property de Palette y suéltelo después de la coma entre las etiquetas `<h2>`. Especifique los siguientes valores en la caja de diálogo de Insert Get Bean Property:
 - **Bean Name:** mybean
 - **Property Name:** name



Figura 3.5. 12. Ventana Insert Bean Property

Haga clic en OK. Note que las etiquetas `<jsp:getProperty>` están entre las etiquetas `<h2>`.

- Haga clic derecho en Source Editor y seleccione Format (Alt-Shift-F) para limpiar el formato de su código. Las etiquetas `<body>` del archivo `response.jsp` deberán ser similares a:

```
<body><jsp:setProperty name="Mybean" property="name" />
    <jsp:getProperty name="Mybean" property="name" />
    <jsp:useBean id="mybean" scope="session" class="org.mypackage.hello.NameI
</body>
```

3.6.6 COMPILANDO Y EJECUTANDO UN PROYECTO DE APLICACIÓN WEB

El IDE usa un script de compilación Ant y ejecuta las aplicaciones Web. El IDE genera el script de compilación basado en las opciones que especifique en el asistente New Project, así como los de la caja de dialogo Project Properties del proyecto (En la ventana Projects, seleccione Properties del menú del nodo del proyecto).

- En la ventana Projects, haga clic derecho sobre el nodo del proyecto HelloWeb y seleccione Run (F6). El IDE compila la aplicación web y la despliega en el servidor que especificó cuando creó el proyecto. La página `index.jsp` se abre en su navegador:



Hello Mundo

Ingrese su nombre:

- Ingrese su nombre en la caja de texto, y haga click en OK. La página response.jsp se mostrará, saludándole:

Nota:

Si compilar y ejecutar el proyecto, se muestra una página de error aparece indicando que response.jsp no está disponible.

Solución de Problema

¿Está utilizando JDK versión 5 o superior? Seleccione Help > del menú principal para verificar la versión del JDK que está usando el IDE. ¿Está usando Tomcat versión 5.x o superior? Seleccione Tools > Servers del menú principal. Seleccione el servidor que el proyecto esta utilizando y observe la entrada Server Type mostrada cerca de la cabecera del diálogo. También puede intentar descargar el proyecto ejemplo y compararlo con el suyo.

3.7 APLICACIÓN WEB MEDIANTE DISEÑO GRÁFICO

Este ejemplo muestra cómo crear una aplicación web simple mediante diseño gráfico, su despliegue en el servidor y su presentación en un navegador.

Para empezar, vamos a crear un proyecto web en nuestro NetBeans:

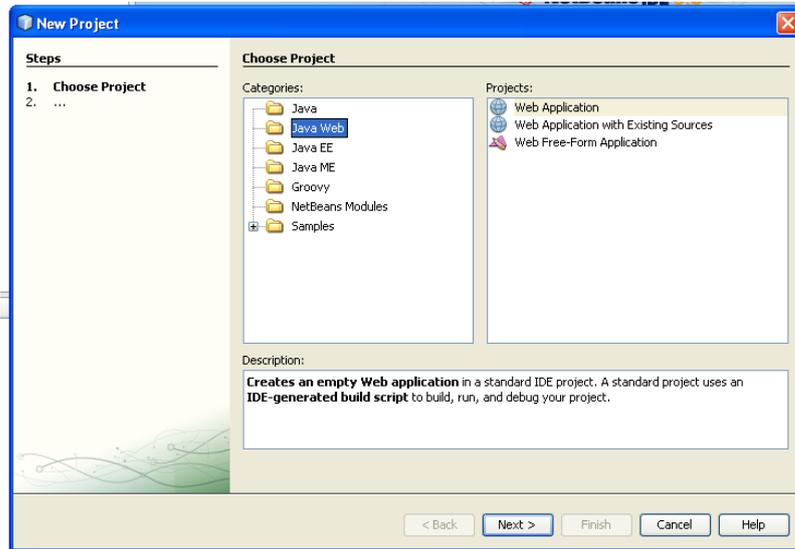


Figura 3.7. 1 New Project Java Web

Introduzca diseño_web en el campo Project Name.

Especifique el project location, cualquier directorio en su computador

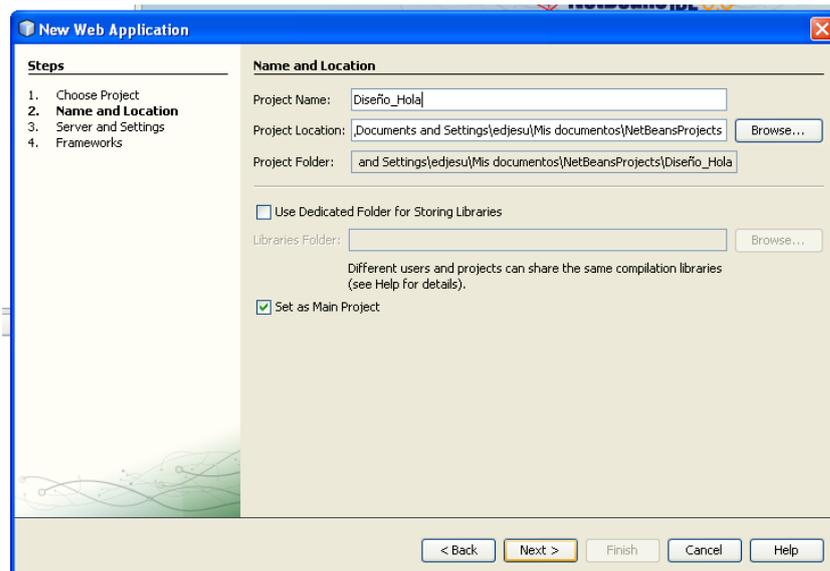


Figura 3.7. 2 Ventana New Web Application

Seleccione el servidor en el cual va a desplegar su aplicación. Solamente se listarán servidores que estén registrados en el IDE. Haga click en next.

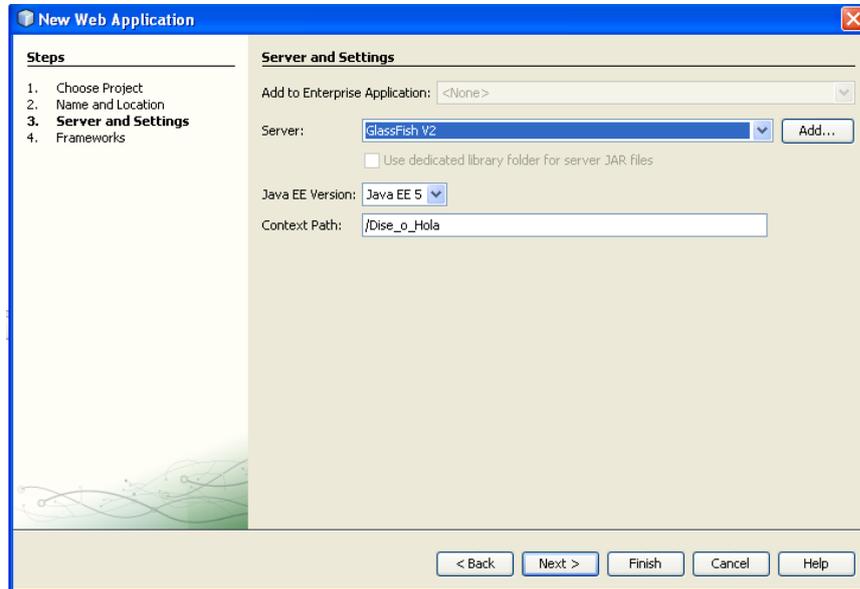


Figura 3.7. 3 Ventana New Web Application Server and Settings

En el panel de Frameworks, seleccione Visual Web JavaServer Faces, haga click en finish para crear el proyecto.

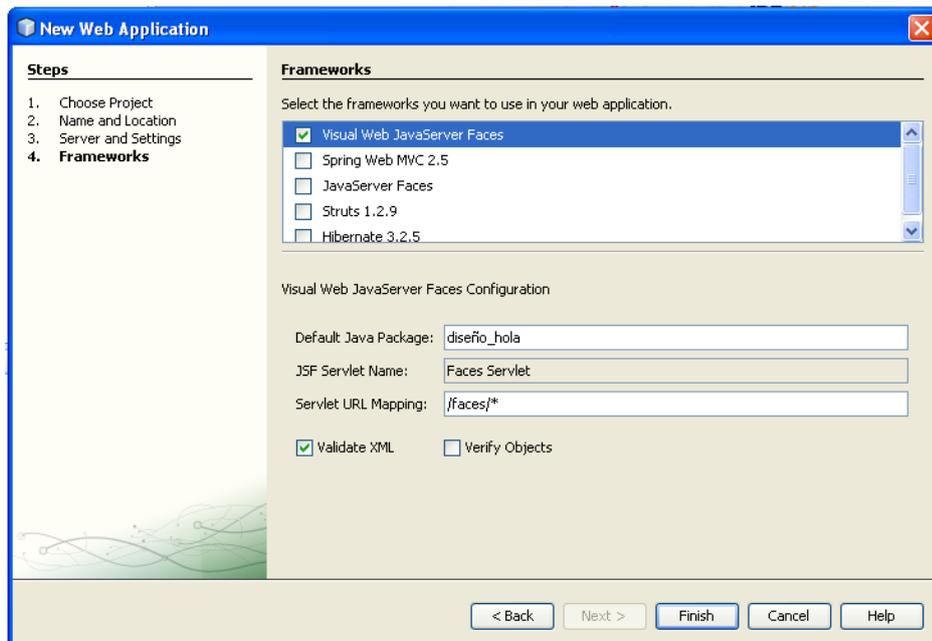


Figura 3.7. 4 Ventana New Web Application Frameworks

La carpeta del proyecto contiene todas las fuentes y metadata del proyecto, como el script de compilación Ant. El proyecto Hola_Web se abre en el IDE. La página de bienvenida, index.jsp, se abre en el Source Editor de la ventana principal. Podrá ver la estructura de archivos del proyecto en la ventana Files (CTRL-2), y su estructura lógica en la ventana Projects (CTRL-1):

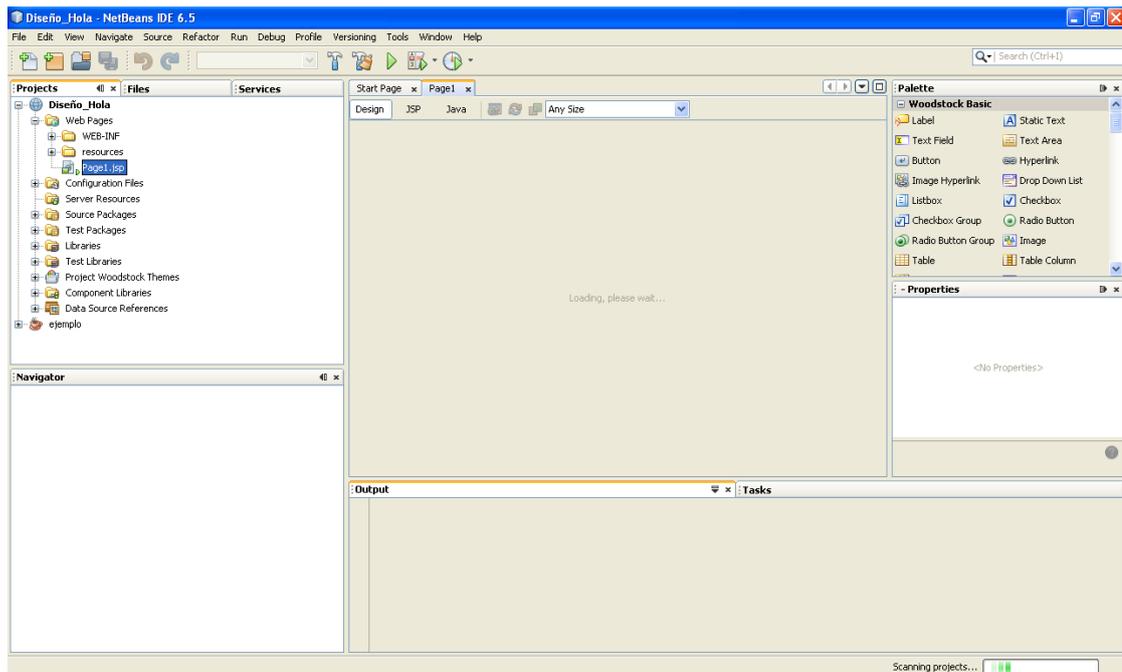


Figura 3.7. 5 Ventana Proyecto Web Diseño_Web

3.7.1 CREANDO Y EDITANDO LOS ARCHIVOS FUENTE DE LA APLICACIÓN WEB

En el diseño grafico escoja de la palette los botones que van a incluir en la aplicación web Diseño_Hola.

- Label
- Text Field
- Button

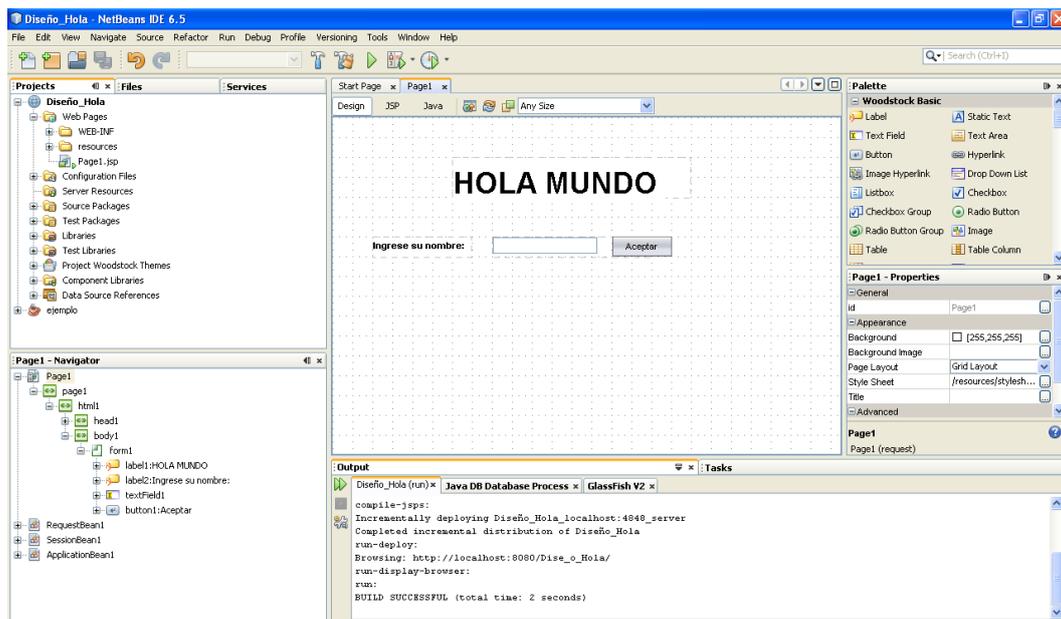
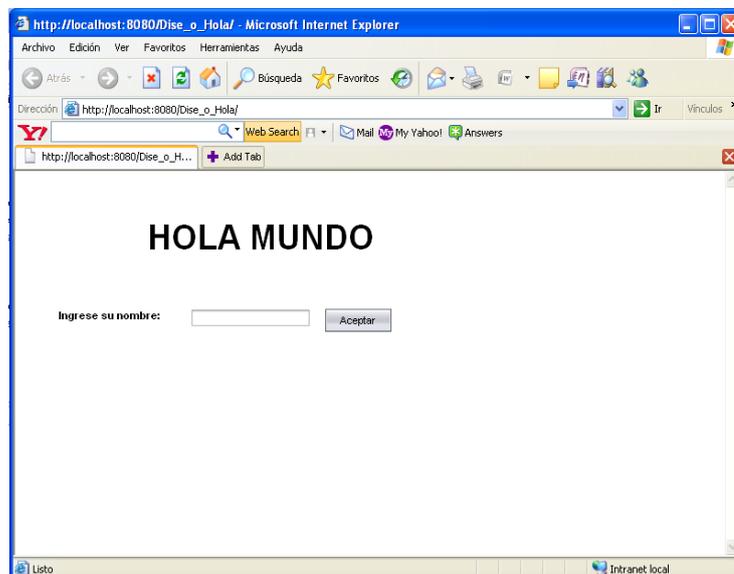


Figura 3.7. 6 Aplicación Web Hola Mundo

Compila y ejecuta el proyecto y este se visualizara en el navegador que tenga disponible.



IV. CAPITULO

4.1 CONCLUSIONES Y RECOMENDACIONES

4.1.1 CONCLUSIONES

- Se instaló NetBeans 6.5 en las todas los sistemas operativos compatibles.
- NetBeans ocupa recursos de la maquina (memoria), por lo que nuestro equipo de trabajo se vuelve lento.
- A pesar que NetBeans 6.5 es mas rápido, se vuelve pesada en maquinas que no cumplen con los requisitos mínimos de hardware.
- NetBeans permite trabajar escribiendo menos código, pues podemos utilizar el autocompletado que dispone esta plataforma.
- La plataforma NetBeans proporciona grandes ventajas en el desarrollo de aplicaciones, pues reduce tiempo, dinero y esfuerzo en la obtención del proyecto final.

4.1.2 RECOMENDACIONES

- Este manual permite que personas con conocimientos elementales de java puedan crear aplicaciones.
- Existen muchas herramientas que son gratuitas por lo que es factible hacer uso de las mismas.
- Promover el uso de software libre, porque existen otras herramientas que pueden acoplarse a las necesidades de nuestro trabajo.
- Día a día debemos ir aprendiendo nuevas técnicas de desarrollo para aprovechar al máximo la productividad que nos puede brindar nuestro equipo, puesto que estas (nuevas técnicas) requieren formación continua
- Si dispone de poca memoria hay otras herramientas que pueden ajustarse a nuestro sistema y a las necesidades de nuestro trabajo.

4.2 BIBLIOGRAFÍA

- www.netbeans.org NetBeans IDE (Entorno Desarrollo de Integrado)
- www.sun.com Sun Microsystems
- www.java.com/es/ Sun Microsystems
- www.java.mexico.com
- www.javahispano.org
- java.com/es/download Página para descargar versiones de java
- todojava.awardspace.com Manual de NetBeans Oficial
- www.ubuntu-es.org Sistema Operativo Ubuntu
- www.ubuntu.com.es
- www.sun.com/software/solaris/ Sun Microsystems SO Solaris