



Introducción a JBuilder®



VERSIÓN 8

Borland®
JBuilder®

Borland Software Corporation
100 Enterprise Way, Scotts Valley, CA 95066-3249
www.borland.com

En el archivo `deploy.html` ubicado en el directorio raíz del producto JBuilder encontrará una lista completa de archivos que se pueden distribuir de acuerdo con la licencia de JBuilder y la limitación de responsabilidad.

Borland Software Corporation puede tener patentes concedidas o en tramitación sobre los temas tratados en este documento. Diríjase al CD del producto o al cuadro de diálogo Acerca de para la lista de patentes. La modificación de este documento no le otorga derechos sobre las licencias de estas patentes.

COPYRIGHT © 1997-2003 Borland Software Corporation. Reservados todos los derechos. Todos los nombres de productos y marcas de Borland son marcas comerciales o registradas de Borland Software Corporation en Estados Unidos y otros países. Las otras marcas pertenecen de sus respectivos propietarios.

Si desea más información acerca de las condiciones de contrato de terceras partes y acerca de la limitación de responsabilidades, consulte las notas de esta versión en su CD de instalación de JBuilder.

Impreso en EE.UU.

JBE0080WW21000introjb 3E2R1002

0203040506-9 8 7 6 5 4 3 2 1

PDF

Índice de materias

Capítulo 1 Introducción

1-1

Capítulo 2 Cómo obtener información sobre JBuilder

2-1

Documentación de JBuilder.	2-1
Convenciones de la documentación	2-4
La ayuda en pantalla de JBuilder.	2-6
Obtención de ayuda	2-6
Principales partes del Visualizador de ayuda	2-7
El Visualizador de ayuda	2-8
El índice de materias	2-8
El índice alfabético	2-8
Búsqueda completa de texto (incluye la búsqueda booleana)	2-8
Búsqueda de texto en el tema actual	2-9
Copia de texto del Visualizador de ayuda.	2-9
Ampliación en el Visualizador de ayuda.	2-9
Utilización de marcadores	2-10
Configuración de los parámetros de proxy	2-10
Inicio del Visualizador de ayuda en una MV independiente	2-11
Desplazamiento por el Visualizador de ayuda.	2-11
Desplazamiento mediante el teclado	2-12
Consulta de la documentación de referencia de las clases	2-12
Diseño de la documentación de referencia.	2-13
Impresión de tutoriales y la documentación	2-13
Recursos adicionales para obtener más información sobre JBuilder	2-13
Asistencia y recursos para desarrolladores	2-14
El servicio de asistencia técnica de Borland	2-14
Recursos en línea	2-14
World Wide Web.	2-15
Grupos de noticias de Borland	2-15
Usenet, grupos de noticias	2-15
Información sobre errores	2-16

Capítulo 3

Tutorial: Creación de una aplicación

3-1

Paso 1: Creación de un proyecto	3-2
Paso 2: Generación de los archivos de código fuente	3-5
Paso 3: Compilación y ejecución de la aplicación	3-9
Paso 4: Personalización de la interfaz del usuario de la aplicación	3-10
Paso 5: Incorporación de componentes a la aplicación	3-13
Paso 6: Modificación del código fuente.	3-15
Paso 7: Compilación y ejecución de la aplicación	3-16
Paso 8: Ejecución de la aplicación desde la línea de comandos	3-16
Paso 9: Asignación de sucesos a botones.	3-17
Paso 10: Cómo completar la interfaz de usuario	3-20
Paso 11: Preparación de la aplicación para la distribución	3-21
Paso 12: Ejecución de la aplicación distribuida desde la línea de comandos	3-24
Código fuente de HolaATodos	3-25
Código fuente de HolaATodosMarco.java	3-25
Código fuente de ClaseHolaATodos.java	3-28
Código fuente de HolaATodosMarco_AboutBox.java	3-29

Capítulo 4

El Visualizador de aplicaciones

4-1

Menú principal	4-3
Barra de herramientas	4-3
Panel del proyecto	4-5
Creación de proyectos	4-6
Adición de archivos y clases a un proyecto	4-6
El panel del proyecto	4-7
El panel de estructura	4-8
Ordenación en el panel de estructura	4-8
Presentación de la información de Javadoc.	4-10
Desplazamiento por el código fuente	4-10
Presentación de mensajes de error	4-11
Panel de contenido	4-12
Pestañas de archivo.	4-12

Pestañas de vistas de archivo.	4-13
Redimensionamiento del panel de contenido .4-13	
El panel de mensajes.	4-13
Copia de texto del panel de mensajes	4-15
Ocultación y presentación del panel de mensajes	4-15
Segregación del panel de mensajes.	4-15
Depurador	4-16
Barras de estado	4-16
Desplazamiento y búsqueda con el Visualizador de aplicaciones	4-17
Búsqueda en árboles.	4-18
Búsqueda de clases	4-18
Utilización de Favoritos.	4-19
Personalización del IDE.	4-20

Capítulo 5

El editor

5-1

Selección de una configuración de teclado para el editor	5-1
Personalización de las definiciones de teclado	5-2
División de la vista del código fuente	5-3
Números de línea	5-4
Presentación de los números de línea	5-4
Desplazamiento a una línea determinada . .	5-5
Selección de texto	5-5
Selección de una línea entera	5-5
Selección de un bloque de números de línea.	5-6
Selección de todos los números de línea. . .	5-6
Desplazamiento de texto mediante la operación de arrastrar y soltar	5-6
Cambio de tamaño de la fuente utilizada en el editor	5-6
Búsqueda de texto en el editor	5-7
Búsqueda de definiciones de símbolos	5-8
Búsqueda de referencias a símbolos	5-8
Cómo aplicar formato al código	5-9
Aplicación de formato al código fuente . .	5-10
Colocación de llaves en los bloques de código	5-10
Métodos abreviados Javadoc del editor	5-10
Utilización de etiquetas @todo en el editor	5-11
Compatibilidad de impresión en el editor . . .	5-12
Arrastrado de archivos al editor	5-13
Atajos de codificación	5-13
ClassInsight.	5-13
MemberInsight	5-15

ClassInsight.	5-15
ParameterInsight.	5-17
Buscar definición (Profundizar)	5-17
Evaluación inmediata de expresiones . . .	5-17
ExpressionInsight	5-17
Configuración de CodeInsight	5-18
Plantillas de código	5-18
Modificar plantillas de código	5-19
Personalización del editor	5-20

Capítulo 6

Automatización del desarrollo de código

6-1

Las herramientas de diseño visual	6-1
La Galería de objetos	6-3
Inicio rápido por medio de los asistentes	6-3

Capítulo 7

Proyectos de JBuilder

7-1

Creación de proyectos	7-2
Adición de archivos a un proyecto	7-4
Presentación de archivos.	7-4
Adición de un archivo fuente Java	7-5
Eliminación y borrado	7-6
Cómo cambiar de nombre proyectos y archivos	7-7
Apertura de proyectos	7-7
Más información sobre los proyectos de JBuilder	7-8

Capítulo 8

Comparación de archivos y versiones

8-1

Glosario de gestión de versiones	8-1
Comparación de dos archivos	8-2
La vista Histórico	8-4
Ficha Contenido	8-6
Ficha Diferencias	8-6
Ficha Información	8-8
Ficha Conflictos en la fusión	8-9

Capítulo 9

Tutorial: Creación de un applet

9-1

Aspectos generales	9-2
El applet Buenas noches	9-3
Paso 1: Creación de un proyecto	9-3
Paso 2: Generación de los archivos de código fuente	9-7

Paso 3: Compilación y ejecución del applet. . .	9-12
Paso 4: Personalización de la interfaz de usuario del applet	9-13
Paso 5: Incorporación de componentes AWT al applet	9-18
Paso 6: Modificación del código fuente	9-21
Paso 7: Distribución del applet.	9-25
Distribución de applets con la herramienta Jar.	9-27
Distribución del applet con el Creador de recopilatorios	9-28
Paso 8: Modificación del archivo HTML	9-32
Paso 9: Ejecución del applet distribuido desde la línea de comandos.	9-34
Comprobación del applet distribuida en la web	9-35
Código fuente del applet	9-36

Código fuente HTML del applet BuenasNochesApplet.html	9-36
Código fuente de la clase del applet BuenasNochesApplet.java	9-37

Capítulo 10	
Tutorial: La vista Histórico	10-1
Paso 1: Generación de varias versiones de un archivo	10-2
Paso 2: La ficha Contenido.	10-4
Paso 3: La ficha Diferencias	10-8
Ordenación de las revisiones.	10-8
Cómo deshacer cambios	10-8
Comparación inteligente	10-10
Paso 4: La ficha Información.	10-11
Índice	I-1

Introducción

Introducción a JBuilder es el punto de partida para los usuarios que no conocen JBuilder. Después de describir brevemente las nuevas características de esta versión de JBuilder, se ofrece una introducción a los recursos disponibles para aprender a utilizar JBuilder y se inicia el trabajo con un tutorial que crea la primera aplicación. Tras esta primera experiencia con el entorno de desarrollo de JBuilder, podrá familiarizarse con el Visualizador de aplicaciones y con el editor y todos los métodos abreviados para procesar el código fuente. Se describen los diversos paneles del Visualizador de aplicaciones y se muestra cómo utilizarlos para ejecutar tareas habituales. Aprenderá a utilizar el editor de JBuilder, incluidos sus diversos métodos abreviados para procesar el código. También aprenderá a crear y gestionar proyectos de JBuilder. Asimismo, aprenderá a comparar archivos y versiones con la vista Histórico de JBuilder. Por último, podrá seguir dos tutoriales básicos adicionales: uno que utiliza la vista Histórico y otro que crea un applet.

Introducción a JBuilder contiene los siguientes capítulos:

- [Capítulo 2, “Cómo obtener información sobre JBuilder”](#)
Describe los recursos que proporciona Borland para ayudarle a conocer mejor el entorno de JBuilder.
- [Capítulo 3, “Tutorial: Creación de una aplicación”](#)
Explica cómo crear una sencilla aplicación “Hola a todos” con JBuilder.
- [Capítulo 4, “El Visualizador de aplicaciones”](#)
Realiza una introducción al Visualizador de aplicaciones y describe la forma de utilizarlo.

- [Capítulo 5, “El editor”](#)
Explica cómo utilizar las numerosas funciones del editor de código de JBuilder, incluidas CodeInsight y las plantillas de código.
- [Capítulo 6, “Automatización del desarrollo de código”](#)
Describe la forma de crear código de forma automática mediante las herramientas de diseño visual y los asistentes de JBuilder.
- [Capítulo 7, “Proyectos de JBuilder”](#)
Explica cómo crear un proyecto de JBuilder, en qué consisten los proyectos y cómo realizar las tareas básicas de gestión de proyectos.
- [Capítulo 8, “Comparación de archivos y versiones”](#)
Aquí se describen las fichas y las funciones disponibles en la ficha Histórico del panel de contenido. Se enumeran las funciones de las distintas versiones de JBuilder, y se explica la forma de trabajar con sistemas de control de versiones y sin ellos.
- [Capítulo 9, “Tutorial: Creación de un applet”](#)
Explica cómo crear un applet AWT.
- [Capítulo 10, “Tutorial: La vista Histórico”](#)
Explica la utilización de la vista Histórico.

Cómo obtener información sobre JBuilder

En este capítulo se describen los recursos disponibles para obtener más información sobre JBuilder. Se describe brevemente el conjunto de la documentación de JBuilder y se explican las convenciones utilizadas en la documentación. Asimismo, se explica cómo utilizar el sistema de ayuda en pantalla de JBuilder; lea este apartado para aprovechar al máximo las funciones que ofrece el sistema de ayuda. También se enumeran otros lugares donde se puede encontrar información adicional sobre el uso de JBuilder. Por último, se indica cómo ponerse en contacto con Borland para solicitar asistencia técnica.

Documentación de JBuilder

Se puede acceder de varias formas a la documentación de JBuilder. Puede ver la documentación en el Visualizador de ayuda de JBuilder mediante la opción de menú Ayuda | Temas de ayuda. Consulte [“La ayuda en pantalla de JBuilder” en la página 2-6](#) para obtener una información completa sobre cómo obtener ayuda en el Visualizador de ayuda. La documentación también se encuentra disponible en el CD de JBuilder, en formato HTML, de modo que se puede utilizar cualquier navegador web para verla.

Si prefiere consultar la documentación en formato impreso, puede adquirir los manuales de JBuilder o descargar los libros en formato PDF e imprimirlos. Para cualquiera de las dos opciones, visite la página web de la documentación de Borland JBuilder 8, en <http://www.borland.com/techpubs/jbuilder>.

Documento	Descripción
Todas las ediciones de JBuilder	
<i>Introducción a JBuilder</i>	Presenta las novedades de esta versión de JBuilder e indica dónde encontrar la información que se busca dentro del conjunto de la documentación de JBuilder. Realiza una rápida introducción a la programación mediante un tutorial que guía en la creación de una sencilla aplicación de Java con JBuilder. Presenta el Visualizador de aplicaciones, el editor de código y algunos de los muchos asistentes y herramientas de producción que permiten realizar el trabajo con mayor rapidez. Explica algunas de las tareas que se realizan habitualmente en la gestión de proyectos de programación. Incluye información y un tutorial sobre la gestión de versiones, y otro tutorial sobre la creación de un applet.
<i>Procedimientos iniciales con Java</i>	Proporciona una descripción general del lenguaje de programación Java, las bibliotecas de clases y las ediciones.
<i>Creación de aplicaciones con JBuilder</i>	Explica cómo gestionar proyectos en JBuilder y cómo utiliza JBuilder las vías de acceso. Explica cómo compilar, depurar, distribuir e internacionalizar programas en JBuilder. Trata las funciones de Javadoc de JBuilder, la presentación y el perfeccionamiento del código, y la comprobación de unidades. También incluye información sobre las herramientas de línea de comandos, los mensajes de error y la forma de trasladar aplicaciones desde otras herramientas de desarrollo en Java.
<i>Diseño de aplicaciones con JBuilder</i>	Explica la forma de diseñar interfaces de usuario y utilizar gestores de diseño. Incluye tutoriales paso a paso para crear interfaces de usuario mediante las herramientas de diseño visual de JBuilder.
<i>Desarrollo de OpenTools para JBuilder (en inglés)</i>	Proporciona información básica sobre la arquitectura de algunos subsistemas de JBuilder y su API OpenTools, con la que se pueden crear extensiones (add-ins) propias.
<i>Referencia de la API de OpenTools (en inglés)</i>	Proporciona documentación de referencia sobre la API de OpenTools de JBuilder.

Documento	Descripción
<i>Documentación del JDK 1.4.1 (en inglés)</i>	<p>Documentación de referencia de la API para el Kit de desarrollo de Java (JDK) de Sun</p> <p>Se puede acceder a esta documentación de dos formas:</p> <ul style="list-style-type: none"> • Elija Ayuda Referencia de Java. • Seleccione la pestaña Doc del panel de contenido con un archivo .JDK abierto.
<i>Especificación del lenguaje Java (en inglés)</i>	Contiene la versión 2.0 de la especificación del lenguaje Java.
<i>Ayuda en pantalla sensible al contexto</i>	Proporciona información relacionada con el elemento actual de la interfaz de JBuilder.
Ediciones SE y Enterprise	
<i>Desarrollo en equipo con JBuilder</i>	Trata la gestión de versiones y explica el funcionamiento de las interfaces de JBuilder con los tres sistemas de control de versiones principales, así como las herramientas de gestión de versiones propias de JBuilder.
Edición Enterprise	
<i>Guía del desarrollador de aplicaciones web</i>	Trata la tecnología que se utiliza para el desarrollo de aplicaciones web en JBuilder.
<i>Guía del desarrollador de servicios web</i>	Explica cómo utilizar las funciones de servicios web de JBuilder para crear, buscar, consumir y publicar servicios web.
<i>Guía del desarrollador de XML</i>	Trata la tecnología que se utiliza para el desarrollo en XML con JBuilder.
<i>Guía del desarrollador de JDataStore</i>	Explica cómo utilizar con eficacia las funciones de JDataStore. JDataStore es una base de datos de altas prestaciones y bajo consumo de recursos, escrita en Java en su totalidad.
<i>Guía del desarrollador de bases de datos</i>	Proporciona información sobre el uso de las funciones de base de datos de DataExpress de JBuilder para el desarrollo de aplicaciones de base de datos.
<i>DataExpress Component Library Reference</i>	Contiene información detallada sobre todos los componentes enlazados a datos, clases, propiedades, métodos y sucesos de valor añadido de JBuilder.
<i>Guía del Desarrollador de Enterprise JavaBeans</i>	Describe la forma de crear Enterprise JavaBeans mediante los diseñadores y las herramientas de EJB de JBuilder.

Convenciones de la documentación

En la documentación de Borland para JBuilder, el texto con significado especial se identifica mediante la tipografía y los símbolos descritos en la siguiente tabla.

Tabla 2.1 Convenciones tipográficas y de símbolos

Tipo de letra	Significado
Letra monoespaciada	<p>El tipo monoespaciado representa lo siguiente:</p> <ul style="list-style-type: none"> • texto tal y como aparece en la pantalla • cualquier cosa que debe escribir, como “Escriba Hola a todos en el campo Título del Asistente para aplicaciones” • nombres de archivos • nombres de vías de acceso • nombres de directorios y carpetas • comandos, como SET PATH • código Java • tipos de datos de Java, como <code>boolean</code>, <code>int</code> y <code>long</code> • identificadores de Java, como nombres de variables, clases, nombres de paquetes, interfaces, componentes, propiedades, métodos y sucesos • nombres de argumentos • nombres de campos • palabras clave de Java, como <code>void</code> y <code>static</code>
Negrita	<p>La negrita se utiliza para las herramientas java, <code>bmj</code> (el Make de Borland para Java), <code>bcj</code> (Compilador de Borland para Java) y opciones del compilador. Por ejemplo: <code>javac</code>, <code>bmj</code>, -vía de acceso a clases.</p>
<i>Cursiva</i>	<p>Las palabras en cursiva indican los términos nuevos que se definen y los títulos de libros; ocasionalmente se usan para indicar énfasis.</p>
<i>Nombres de tecla</i>	<p>Este tipo de letra indica una tecla, como “Pulse <i>Esc</i> para salir de un menú”.</p>
[]	<p>Los corchetes, en las listas de texto o sintaxis, encierran elementos optativos. En estos casos no se deben escribir los corchetes.</p>

Tabla 2.1 Convenciones tipográficas y de símbolos (continuación)

Tipo de letra	Significado
< >	<p>Los corchetes angulares se utilizan para indicar las variables en vías de acceso a directorios, opciones de comandos y ejemplos de código.</p> <p>Por ejemplo, <nombredearchivo> se puede utilizar para indicar que es necesario especificar un nombre de archivo (incluida su extensión); <nombredeusuario> indica que se debe escribir un nombre de usuario.</p> <p>Cuando reemplace las variables en vías de acceso a directorios, opciones de comandos y ejemplos de código, sustituya la variable entera, incluidos los corchetes angulares (< >). Por ejemplo, sustituya <nombredearchivo> por el nombre de un archivo, como por ejemplo <code>empl eado.js</code>, y quite los corchetes angulares.</p> <p>Nota: Los archivos HTML, XML, JSP y de otros formatos basados en etiquetas utilizan también corchetes angulares para delimitar elementos del documento, como por ejemplo: y <ejb-jar>. La siguiente convención describe la forma de especificar cadenas de variables dentro de los ejemplos de código en cuya sintaxis se utilizan corchetes angulares como delimitadores.</p>
<i>Cursiva, serif</i>	<p>Este formato se utiliza para indicar las cadenas de variables en los ejemplos de código que ya están usando corchetes como delimitadores. Por ejemplo, <url="jdbc:borl and:jbuilder\\samples\\guestbook.js"></p>
...	<p>En los ejemplos de código, los puntos suspensivos (...) indican código que se ha omitido en el ejemplo para ahorrar espacio y aumentar la claridad. Si están en un botón, los puntos suspensivos indican que éste conduce a un cuadro de diálogo de selección.</p>

JBuilder se puede utilizar con diversas plataformas. Consulte la siguiente tabla para ver una descripción de las convenciones de plataforma utilizadas en la documentación.

Tabla 2.2 Convenciones de las plataformas

Elementos	Significado
Vías de acceso	<p>Las vías de acceso a los directorios en la documentación se indican con una barra normal (/).</p> <p>Para la plataforma Windows se utiliza la barra invertida (\).</p>

Tabla 2.2 Convenciones de las plataformas (continuación)

Elementos	Significado
Directorio de inicio	<p>La ubicación del directorio inicial varía según la plataforma y se indica con una variable <home>.</p> <ul style="list-style-type: none"> • En UNIX y Linux, el directorio inicial puede variar. Por ejemplo, puede ser /user/<nombre de usuario> o /home/<nombre de usuario> • En Windows NT, el directorio inicial es C:\Winnt\Profiles\<nombre de usuario> • En Windows 2000 y XP, el directorio inicial es C:\Documents and Settings\<nombre de usuario>
Imágenes de pantalla	Las imágenes o capturas de pantalla utilizan el aspecto Metal en diversas plataformas.

La ayuda en pantalla de JBuilder

JBuilder presenta los temas de ayuda en pantalla en el Visualizador de ayuda. Los temas de ayuda también se pueden consultar en el Visualizador de aplicaciones y en un navegador.

Obtención de ayuda

Existen varias formas de obtener ayuda sobre cualquier tema:

- Para obtener ayuda desde el IDE:
 - Para abrir el Visualizador de ayuda, elija Temas de ayuda en el menú Ayuda de JBuilder.
 - Pulse el botón Ayuda que aparece en los cuadros de diálogo o *F1*.
 - Seleccione Buscar | Buscar clases y elija una clase. El archivo de clase se abre en el panel de contenido. Pulse la pestaña Doc para ver si existe documentación disponible para esa clase.
- Para obtener ayuda desde el AppBrowser:
 - Haga doble clic en un nombre de clase, en el panel de estructura o del proyecto, y seleccione la pestaña Doc. Aparece la documentación de esta clase.
 - Seleccione un nombre de clase en el panel de estructura o del proyecto y pulse *Intro*; a continuación, abra la pestaña Doc. Equivale a hacer doble clic.
 - Sitúe el cursor sobre una palabra clave de Java, en el editor, y pulse *F1* para obtener ayuda.
- Desde el Inspector, seleccione una propiedad o un suceso y pulse *F1*.

- Si desea obtener ayuda desde el navegador web, siga estos pasos:
 - a Extraiga el contenido de todos los recopilatorios `.jar` del directorio `doc` de JBuilder. Para ello puede emplear la herramienta que desee, como WinZip, o bien, desde la línea de comandos, sitúese en el directorio `doc` y escriba lo siguiente por cada archivo JAR:






```
../<jdk>/bin/jar xf <doc>jarfile.jar.
```
 - b Después de extraer todos los archivos JAR, abra `index.html` con el navegador y siga las instrucciones.

Nota No es posible realizar búsquedas completas de texto en la documentación en pantalla cuando se accede a ella mediante un navegador.

Principales partes del Visualizador de ayuda

El Visualizador de ayuda de JBuilder incluye lo siguiente:

- El menú principal.
- La ficha Índice de materias, que muestra un índice de materias desplegable de todos los manuales.
- La ficha Índice, que muestra las entradas de índice de todos los manuales.
- La ficha Buscar, que permite introducir palabras para buscarlas en todos los manuales.
- El panel de contenido, que muestra el texto del tema seleccionado.
- Los siguientes botones:

Botón	Nombre	Descripción
	Inicio	Va al primer tema de la lista.
	Atrás	Va al tema anterior de la lista.
	Adelante	Va al tema siguiente de la lista.
	Imprimir	Imprime el tema actual.
	Buscar	Busca texto en el tema actual.

El Visualizador de ayuda

El índice de materias

Para elegir un tema de ayuda del índice de materias:

- 1 Haga clic en la pestaña Índice materias, en la esquina superior izquierda del Visualizador de ayuda.
- 2 Haga doble clic en el icono de un manual para ampliar sus temas.
- 3 Seleccione un elemento del índice de materias para consultar el tema en el panel de contenidos.

El índice alfabético

Para buscar un tema en el índice:

- 1 Haga clic en la pestaña Índice, situada en la parte superior izquierda del Visualizador de ayuda.
- 2 Escriba el tema en el campo de texto.
A medida que se escribe, el índice alfabético se desplaza y realiza una búsqueda gradual entre las entradas.
- 3 Para mostrar el tema resaltado en el índice, haga doble clic en él o pulse *Intro*.

Si hay más de un tema para la entrada de índice seleccionada, el panel del índice se divide en dos y muestra una lista de temas en la parte inferior. Pulse el tema deseado.

Búsqueda completa de texto (incluye la búsqueda booleana)

Para buscar texto en todos los manuales y utilizar la búsqueda booleana:

- 1 Haga clic en la pestaña Buscar, situada en la esquina superior izquierda del Visualizador de ayuda.
- 2 Escriba en el campo de texto las palabras que desea buscar.

A medida que se escribe, la búsqueda se desplaza hacia la correspondencia más cercana.

El Visualizador de ayuda admite las siguientes operaciones de búsqueda booleanas (no utilice comillas en el campo de texto):

- Intersección de temas (AND):

Escriba un signo más (+) entre las palabras si desea buscar los temas que incluyan todas esas palabras.

- Unión de temas (OR):

Escriba una coma (,) entre las palabras si desea buscar todos los temas que incluyan al menos una de esas palabras.

- 3 Para ver el contenido del tema resaltado, haga doble clic en él o pulse *Intro*.

Si hay más de un tema para la entrada de índice seleccionada, el panel Buscar se divide en dos y muestra una lista de temas en la parte inferior. Pulse el tema deseado.

Sugerencia La búsqueda completa de texto devuelve resultados de todo el texto en todos los manuales de la documentación en pantalla. Si devuelve demasiados resultados, pruebe a utilizar en su lugar la pestaña Índice.

Búsqueda de texto en el tema actual

Busca texto en el tema actual del visualizador de ayuda.



- 1 Pulse el botón Buscar del Visualizador de ayuda o pulse *Ctrl + F*.
- 2 Escriba las palabras en el cuadro de diálogo Buscar en el tema actual.
- 3 Pulse Buscar siguiente o *Intro* para continuar con la búsqueda.
- 4 Pulse Cancelar para cerrar el cuadro de diálogo.
- 5 Pulse *F3* para buscar de nuevo sin abrir el cuadro de diálogo.

Copia de texto del Visualizador de ayuda

Para copiar texto del Visualizador de ayuda

- 1 Seleccione el texto en el panel de contenido.
- 2 Seleccione Edición | Copiar en el menú principal del Visualizador de ayuda o pulse *Ctrl + c*.

Ampliación en el Visualizador de ayuda

Es posible ampliar y reducir el tamaño de la fuente y las imágenes que se muestran en el Visualizador de ayuda. Así puede examinar las imágenes con más precisión, hacer que el texto encaje en la vista o adaptar el tamaño de presentación a su gusto. Los incrementos son pequeños, lo que permite controlar con precisión el tamaño de la vista.

Existen dos formas de ampliar o reducir la vista:

Para ampliar:

- Pulse *Ctrl + U*, mantenga pulsada la tecla *Ctrl* y pulse *U* varias veces hasta que la vista tenga el tamaño deseado.
- Seleccione Opciones | Ampliar.

Para reducir:

- Pulse *Ctrl + D*; mantenga pulsada la tecla *Ctrl* y pulse *D* varias veces hasta que la vista tenga el tamaño deseado.
- Seleccione Opciones | Reducir.

Para volver a la vista normal, seleccione Opciones | Zoom normal.

Utilización de marcadores

Si desea añadir un marcador a la ficha actual del Visualizador de ayuda, seleccione Marcadores | Añadir marcador en el menú principal.

Para modificar marcadores en el Visualizador de ayuda:

- 1 Seleccione Marcadores | Modificar marcadores en el menú principal. Se abre el cuadro de diálogo Modificar marcadores.
- 2 Seleccione el marcador y haga clic en la opción deseada para eliminar el marcador, reorganizarlo o desplazarse a él.
- 3 Pulse Cerrar.

Para ir a un marcador, seleccione Marcadores y elija un marcador de la lista. Los primeros diez marcadores disponen de atajos en el menú Marcadores, por lo que es preferible colocar los más utilizados al principio de la lista. Existe un límite de 25 marcadores en el menú, aunque este número se puede aumentar modificando el archivo `help.properties`. El archivo `<help.properties>` se crea la primera vez que se sale del Visualizador de ayuda, y se guarda en el subdirectorio `.jbuilder` del directorio inicial.

Configuración de los parámetros de proxy

Para poder acceder a Internet desde el Visualizador de ayuda a través de un servidor proxy, debe especificar el ID del anfitrión y el número de puerto del servidor proxy. Estos parámetros se configuran en el Visualizador de ayuda.

Para hacer que el Visualizador de ayuda utilice un servidor proxy para conectarse a Internet:

- 1 Abra el Visualizador de ayuda (Ayuda | Temas de ayuda).
- 2 Seleccione Opciones | Más para abrir el cuadro de diálogo Opciones del sistema de ayuda.
- 3 Active la opción Conectarse a Internet mediante servidor proxy.
- 4 En el campo Host, escriba el nombre o la dirección IP numérica del servidor proxy.
- 5 En el campo Puerto, escriba el número de puerto del servidor proxy.
- 6 Pulse Aceptar.
- 7 Reinicie JBuilder para que se aplique el cambio.

Inicio del Visualizador de ayuda en una MV independiente





Se puede ejecutar el Visualizador de ayuda en una máquina virtual (MV) independiente, de modo que se pueda acceder a los tutoriales y a otros temas de ayuda mientras se muestra cuadro de diálogo modal. Esta opción se configura en el Visualizador de ayuda.

Para configurar el Visualizador de ayuda de modo que se inicie en una MV independiente:

- 1 Abra el Visualizador de ayuda (Ayuda | Temas de ayuda).
- 2 Seleccione Opciones | Más para abrir el cuadro de diálogo Opciones del sistema de ayuda.
- 3 Active la opción Iniciar sistema de ayuda en MV distinta.
- 4 Pulse Aceptar.
- 5 Reinicie JBuilder para que se aplique el cambio.

Desplazamiento por el Visualizador de ayuda

En la tabla siguiente se describe la forma de desplazarse por el Visualizador de ayuda:

Para	Haga esto	Métodos abreviados de teclado
Cambiar de manual o ir a un tema	Pulse el título del tema.	Utilice las teclas de flecha.
Ampliar o contraer una sección del índice de materias	Pulse el icono contiguo a un tema en la ficha Índice de materias o haga doble clic en el tema.	Seleccione el manual y pulse <i>Intro</i> para abrirlo. También puede seleccionar el manual y utilizar las teclas de flecha derecha e izquierda para abrirlo y cerrarlo.
Volver al tema anterior de la lista de historial	 Pulse el botón Atrás.	<i>Ctrl+flecha izquierda</i>
Ir al tema siguiente de la lista	 Pulse el botón Adelante.	<i>Ctrl+flecha derecha</i>
Ir al primer tema de la lista	 Pulse el botón Inicio.	<i>Ctrl+Inicio</i>
Buscar texto en el tema actual	 Pulse el botón Buscar.	<i>Ctrl+F</i>
Buscar de nuevo		<i>F3</i>

Para	Haga esto	Métodos abreviados de teclado
Buscar texto en todos los manuales	Haga clic en la pestaña Buscar.	
Seguir un enlace de hipertexto subrayado en el texto	Haga clic en el enlace del texto.	

Desplazamiento mediante el teclado

Cuando se selecciona una entrada de la ficha Índice de materias, Índice o Buscar del Visualizador de ayuda, las acciones del teclado tienen efecto sobre el lado izquierdo. Las teclas de desplazamiento (como Av Pág) desplazan la parte izquierda del Visualizador de ayuda, no el panel de contenido de la parte derecha.

Para pasar el foco del teclado al panel de contenido, situado en el lado derecho del Visualizador de ayuda, pulse la tecla Tab. El tabulador pasa del panel seleccionado en el lado izquierdo del Visualizador de ayuda (Contenido, Índice o Buscar) al panel de contenido y después a los botones de desplazamiento. Pulse *Mayús+Tab* para retroceder.

Consulta de la documentación de referencia de las clases

La documentación de referencia de las clases se puede consultar en el Visualizador de aplicaciones, en el Visualizador de ayuda o con un navegador.

Los siguientes métodos permiten presentar la documentación en el Visualizador de aplicaciones:

- Haga doble clic en una clase del panel de estructura o del proyecto para ver su código fuente en el panel del código fuente. Abra la pestaña Doc para ver la documentación correspondiente.
- Elija Buscar | Buscar clases y seleccione una clase. Abra la pestaña Doc para ver la documentación correspondiente.

Para que sea posible encontrar la documentación de la clase, las sentencias `import` de la parte superior del archivo de código fuente deben apuntar a la clase deseada. Si aparece el mensaje No se encuentra el símbolo Java, elija un archivo fuente que importe la clase deseada.

Para presentar las guías de referencia en el Visualizador de ayuda, utilice uno de los siguientes métodos:

- En la ficha Atajos del Visualizador de ayuda, haga clic en un manual de referencia.
- En la ventana principal de JBuilder, seleccione Ayuda y la guía deseada.

- En la ficha Índice de materias del Visualizador de ayuda, haga clic en el enlace de un manual de referencia.

Diseño de la documentación de referencia

La documentación en la *DataExpress Component Library Reference* se envía en formato Javadoc estándar. Si desea información sobre el uso del formato Javadoc, consulte *This API Document Is Organized* en *DataExpress Component Library Reference*.

Impresión de tutoriales y la documentación

Para imprimir los tutoriales y otra documentación de JBuilder, abra el documento deseado en el Visualizador de ayuda y elija Archivo | Imprimir. La documentación también está disponible en formato PDF en el CD de JBuilder y en la página web de JBuilder, en <http://www.borland.com/techpubs/jbuilder/>.

Si desea consultar una lista de los tutoriales y los nombres de sus archivos PDF, elija Ayuda | Tutoriales de JBuilder.

Recursos adicionales para obtener más información sobre JBuilder

Si desea obtener más información sobre JBuilder, consulte estas opciones:

- La página web de JBuilder, que se encuentra en <http://www.borland.com/jbuilder/>, y la página de la Comunidad Borland, en <http://community.borland.com/>.
- Si desea una introducción general a JBuilder, abra el Proyecto de bienvenida. Seleccione Archivo | Bienvenida en el menú principal de JBuilder.
- Consulte las notas de la versión (Ayuda | Notas de esta versión) si desea información sobre las últimas actualizaciones de JBuilder, así como los problemas conocidos y sus soluciones sugeridas.

Asistencia y recursos para desarrolladores

Borland proporciona una serie de opciones de asistencia y recursos de información para ayudar a los desarrolladores a obtener el máximo rendimiento de los productos Borland. Entre estas opciones se incluye una gama de programas de asistencia técnica de Borland, así como servicios gratuitos en Internet, los cuales permiten consultar una amplia base de información y ponerse en contacto con otros usuarios de productos Borland.

El servicio de asistencia técnica de Borland

Borland ofrece varios programas de asistencia para clientes actuales y potenciales. Se puede elegir entre varios tipos de asistencia, que van desde la ayuda en la instalación de los productos Borland hasta el asesoramiento de expertos y la asistencia pormenorizada.

Si desea más información sobre el servicio al desarrollador de Borland, visite nuestra página web, en <http://www.borland.com/devsupport>.

Cuando se ponga en contacto con el servicio técnico tenga a mano la información completa sobre el entorno, la versión del producto utilizada y una descripción detallada del problema.

Si necesita más información sobre las herramientas o la documentación de otros proveedores, póngase en contacto con ellos.

Recursos en línea

También puede obtener información de los siguientes recursos en línea:

World Wide Web	http://www.borland.com/
FTP	ftp://ftp.borland.com/ Documentación técnica disponible por ftp anónimo.
Listserv	Para suscribirse a las circulares electrónicas, rellene el formulario en línea que aparece en: http://info.borland.com/contact/listserv.html y para el servidor de listas internacional Borland: http://info.borland.com/contact/intl-list.html

World Wide Web

Visite periódicamente www.borland.com/jbuilder. El equipo de desarrollo de productos Java publica en esta página documentación técnica, análisis de competitividad, respuestas a preguntas frecuentes, aplicaciones de ejemplo, software actualizado e información sobre productos nuevos y antiguos.

En particular, pueden resultar interesantes las siguientes direcciones:

- <http://www.borland.com/jbuilder/> (actualizaciones de software y otros archivos)
- <http://www.borland.com/techpubs/jbuilder/> (actualizaciones de documentación y otros archivos)
- <http://community.borland.com/> (contiene nuestra revista de noticias para desarrolladores en formato web)

Grupos de noticias de Borland

Puede registrar JBuilder y participar en los grupos de debate sobre JBuilder, estructurados en hilos. Los grupos de noticias de Borland ofrecen un medio para que la comunidad internacional de clientes de Borland pueda intercambiar sugerencias y técnicas sobre los productos de Borland, así como las herramientas y tecnologías relacionadas.

Puede encontrar grupos de noticias, moderados por los usuarios, sobre JBuilder y otros productos de Borland, en <http://www.borland.com/newsgroups>.

Usenet, grupos de noticias

En Usenet existen los siguientes grupos dedicados a Java y temas relacionados:

- news:comp.lang.java.advocacy
- news:comp.lang.java.announce
- news:comp.lang.java.beans
- news:comp.lang.java.databases
- news:comp.lang.java.gui
- news:comp.lang.java.help
- news:comp.lang.java.machine
- news:comp.lang.java.programmer
- news:comp.lang.java.security
- news:comp.lang.java.softwaretools

Nota Se trata de grupos moderados por usuarios; no son páginas oficiales de Borland.

Información sobre errores

Si encuentra algún error en el software, comuníquelo en la página Support Programs, en <http://www.borland.com/devsupport/namerica/>. Pulse el enlace “Reporting Defects” para llegar al formulario Entry.

Cuando informe sobre un fallo, incluya todos los pasos necesarios para llegar a él, así como toda la información posible sobre la configuración, el entorno y las aplicaciones que se estaban utilizando junto con JBuilder. Intente explicar con la mayor claridad posible las diferencias entre el comportamiento esperado y el obtenido.

Si desea enviar felicitaciones, sugerencias o quejas al equipo de documentación de JBuilder, envíe un mensaje a jgpubs@borland.com. Envíe únicamente comentarios sobre la documentación. Tenga en cuenta que los asuntos relacionados con el servicio técnico se deben enviar al departamento de asistencia técnica para programadores.

JBuilder es una herramienta creada por desarrolladores y para desarrolladores. Valoramos sumamente sus aportaciones.

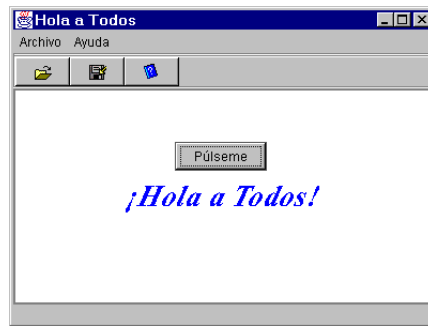
Tutorial: Creación de una aplicación

Muchos desarrolladores prefieren aprender con la práctica. Si ése es su caso, los pasos de este tutorial le enseñarán a utilizar el entorno integrado de desarrollo (IDE) de JBuilder para crear aplicaciones. El tutorial muestra cómo:

- Crear un proyecto para la aplicación.
- Crear una sencilla aplicación “Hola a todos”.
- Compilar y ejecutar la aplicación.
- Modificar la interfaz de usuario de una aplicación.
- Añadir componentes Swing, tales como JPanel, JLabel y JButton.
- Modificar el código fuente.
- Ejecutar la aplicación desde la línea de comandos.
- Asociar un método de suceso a un botón.
- Completar la interfaz de usuario.
- Agrupar los archivos para publicación.
- Ejecutar la aplicación distribuida desde la línea de comandos.

Son funciones de
JBuilder SE y Enterprise

Cuando termine el tutorial tendrá una aplicación en Java con el siguiente aspecto:



Si necesita información concreta acerca del IDE o el editor cuando trabaja en el tutorial, consulte el [Capítulo 4, “El Visualizador de aplicaciones”](#) y el [Capítulo 5, “El editor”](#).

Consulte el apartado Tutoriales en las Sugerencias de JBuilder si desea información sobre cómo ver e imprimir tutoriales. El apartado Opciones de accesibilidad de Sugerencias de JBuilder contiene sugerencias sobre la utilización de las funciones de JBuilder para mejorar la facilidad de uso de JBuilder para personas con discapacidades.

Para obtener información sobre las convenciones utilizadas en este tutorial y en otra documentación de JBuilder, consulte [“Convenciones de la documentación” en la página 2-4](#).

Paso 1: Creación de un proyecto

Antes de crear una aplicación con JBuilder es necesario crear un proyecto para trabajar en él. JBuilder utiliza un archivo de proyecto con extensión `.jpx` para organizar los archivos de aplicación y mantener la configuración y las propiedades del proyecto. El asistente para proyectos crea los proyectos automáticamente.

- 1 Elija Archivo | Nuevo proyecto para iniciar el Asistente para proyectos.
- 2 En el primer paso del Asistente para proyectos, realice los siguientes cambios en los campos correspondientes:
 - a Escriba `HolaATodos` en el campo Nombre.

Nota

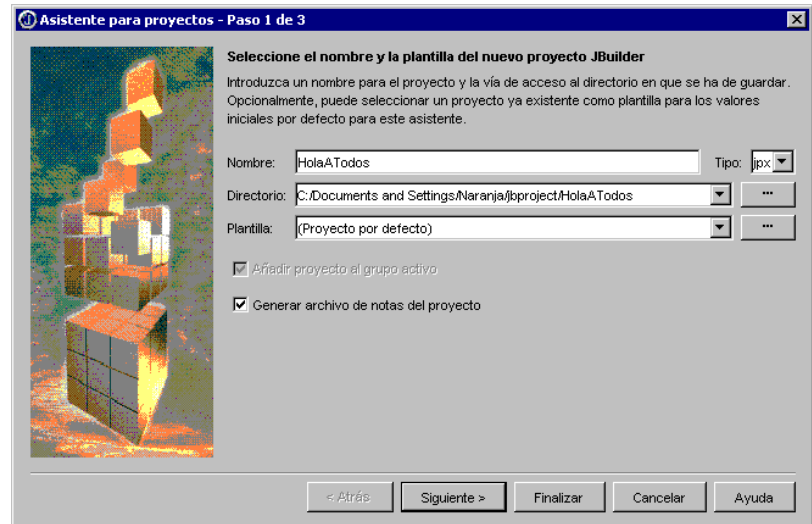
Cuando se escribe el nombre del proyecto en el campo Nombre, se escribe el mismo nombre en el campo Directorio. Por defecto, JBuilder utiliza este nombre de proyecto para crear los nombres del directorio del proyecto y el paquete de las clases. El nombre del paquete se convierte en minúsculas, según la convención de nomenclatura de paquetes de Java. Por defecto, los proyectos de JBuilder se guardan en el directorio `/inicio/jbproject`. El directorio

inicial varía según la plataforma. Consulte [“Convenciones de la documentación” en la página 2-4](#) y si desea obtener más información sobre los proyectos, consulte el [Capítulo 7, “Proyectos de JBuilder”](#).

- b Acepte `jspx` como tipo de archivo del proyecto.
- c Seleccione la opción Generar archivo de notas del proyecto.

Cuando se activa esta opción, el Asistente para proyectos crea un archivo HTML para las notas del proyecto y lo añade.

El primer paso del Asistente ofrecerá un aspecto semejante a este:



- 3 Acepte todas las demás opciones por defecto del Paso 1.
- 4 Pulse Siguiente para ir al Paso 2 del Asistente para proyectos.
- 5 Acepte las vías de acceso por defecto del Paso 2. Observe dónde se guardan los archivos de clase compilados, los archivos de proyecto y los archivos de código fuente.

Si desea más detalles sobre las vías de acceso y cómo guarda JBuilder los archivos, consulte *“Gestión de las vías de acceso” en Creación de aplicaciones con JBuilder*.

El segundo paso del Asistente ofrecerá un aspecto semejante a este:

6 Haga clic en Siguiente para avanzar al Paso 3 del asistente.

7 Escriba lo siguiente en los campos apropiados del Paso 3:

- a Acepte las opciones por defecto para Codificación y Recopilación automática de paquetes fuente. La recopilación automática de paquetes fuente es una función de JBuilder SE y Enterprise.
- b Escriba *Hola A Todos* en el campo Título de los campos de clase Javadoc.
- c Escriba su nombre, el nombre de la empresa y una descripción de la aplicación en los campos optativos correspondientes.

Nota

La información de los campos de clase Javadoc aparece en el archivo del proyecto HTML y en forma de comentarios de cabecera optativos en el código fuente.

El tercer paso del asistente para proyectos tiene el siguiente aspecto:

Asistente para proyectos - Paso 3 de 3

Especificar configuración general del proyecto

Introduzca parámetros aquí para ayudar a definir su nuevo proyecto. Estas y otras propiedades pueden cambiarse después de crear el proyecto.

Codificación: Por defecto

☐ Activar assert como palabra clave

Recopilación automática de paquetes fuente

☒ Activar la localización y compilación de paquetes fuente

Niveles jerárquicos mostrados: 3

Archivos Javadoc de clases:

Etiqueta	Texto
Título:	HolaATodos
Descripción:	Aplicación inicial
Copyright:	Copyright (c) 2002
Empresa:	
@author	
@version	1.0

☐ Incluir referencias de archivos de clase de las bibliotecas del proyecto

< Atrás Siguiente > Finalizar Cancelar Ayuda

- 8 Pulse el botón Finalizar.

En el panel del proyecto del Visualizador de aplicaciones aparecen dos archivos generados por el asistente, `HolaATodos.jpx` y `HolaATodos.html`.

- 9 Haga doble clic en `HolaATodos.html`, el archivo de notas del proyecto, para verlo en el panel de contenido del centro del Visualizador de aplicaciones. Tenga en cuenta que contiene el nombre de proyecto, el autor, la empresa y la información descriptiva introducida en el Paso 3 del asistente para proyectos.

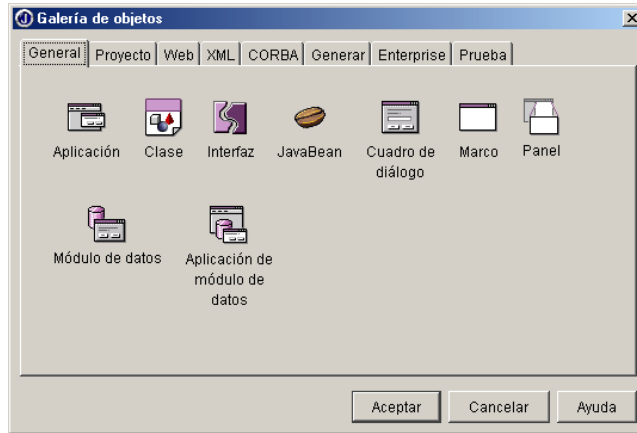
Paso 2: Generación de los archivos de código fuente

El Asistente para aplicaciones crea archivos de código fuente `.java` que se añaden al proyecto que se acaba de crear.

Para generar los archivos fuente de una aplicación con el Asistente para aplicaciones, siga estas instrucciones:



- 1 Elija Archivo | Nuevo o pulse el botón Nuevo de la barra de herramientas principal para abrir la Galería de objetos.



- 2 Seleccione la pestaña General y haga doble clic en el icono Aplicación para que se abra el Asistente para aplicaciones.
- 3 Acepte el nombre de paquete por defecto, `hol.aatodos`, en la primera ficha del asistente para aplicaciones.

Por defecto, el asistente toma el nombre del paquete del nombre del archivo del proyecto, `Hol.aATodos.jp`.

- 4 Escriba `ClaseHol.aATodos` en el campo Clase.

Se trata de un nombre de clase de Java que distingue entre mayúsculas y minúsculas.

- 5 Seleccione la opción Generar comentarios de cabecera.

Esta opción hace que la información sobre notas de proyecto introducida en el Paso 3 del asistente para proyectos aparezca al principio de los archivos fuente que genera el asistente para aplicaciones.

El primer paso del Asistente para aplicaciones ofrecerá el siguiente aspecto:



6 Pulse el botón Siguiente para ir al Paso 2 del Asistente para aplicaciones.

7 Escriba MarcoHolaATodos en el campo Clase.

8 Escriba Hola a todos en el cuadro Título.

Este texto aparece en la barra de título del marco de la aplicación.

9 Seleccione todas las opciones para activar las funciones adicionales de la aplicación: Generar barra de menú, Generar barra de herramientas, Generar barra de estado, Generar cuadro Acerca de y Centrar marco en la pantalla. El asistente genera el código básico para crear estas funciones.

El segundo paso del Asistente para aplicaciones tiene el siguiente aspecto:



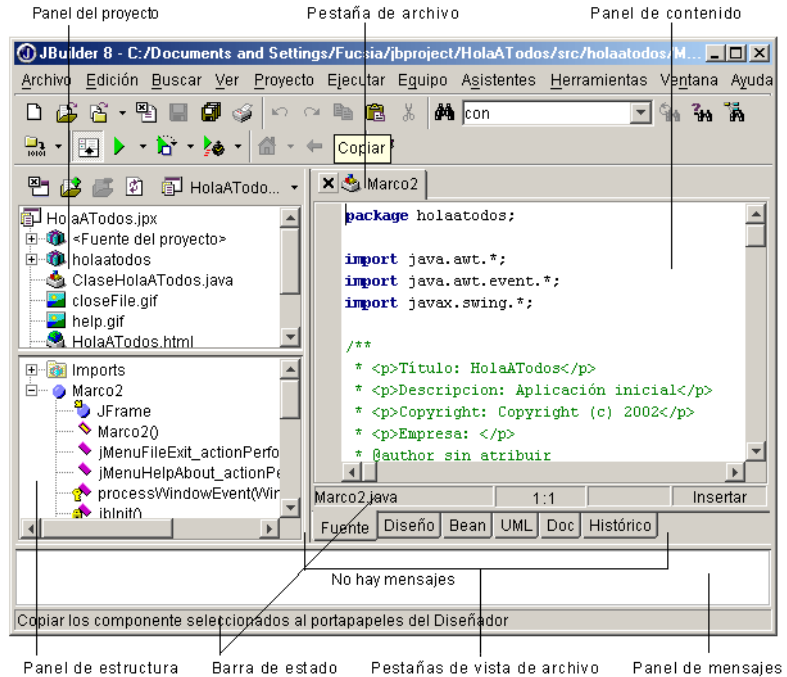
10 Pulse el botón Finalizar.

Los nuevos archivos de código fuente `.java` y las imágenes de la barra de herramientas se añaden al proyecto y se muestran como nodos en el panel del proyecto. El código fuente de `HolaATodosMarco.java` se abre en el panel de contenido, como se muestra en la imagen siguiente.

Nota El panel de mensajes sólo aparece cuando se muestra un mensaje. Por lo tanto, aún no verá el panel de mensajes si está siguiendo estos pasos.

Nota En las ediciones de JBuilder SE y Enterprise también aparece automáticamente un nodo de paquete de código fuente llamado `HolaATodos` si se ha elegido la opción Activar la localización y compilación de paquetes de código fuente de la ficha General del cuadro de diálogo Propiedades de proyecto (Proyecto | Propiedades de proyecto).

Figura 3.1 Elementos del visualizador de aplicaciones



- 11 Seleccione Archivo | Guardar todo para guardar los archivos de código fuente y el archivo de proyecto.

Nota

Los archivos fuente se guardan en:

/<inicio>/j bproject/Hol aATodos/src/hol aatodos

Los archivos de clase generados por el compilador Java a partir de los archivos fuente se guardan en:

/<inicio>/j bproject/Hol aATodos/clases/hol aatodos

Paso 3: Compilación y ejecución de la aplicación

Ahora, compile y ejecute la aplicación. La compilación es el proceso de ejecutar el compilador Java. El compilador, que convierte el código fuente en bytecode de Java, genera archivos .class.



Sugerencia

- 1 Elija Ejecutar | Ejecutar el proyecto, o haga clic en el botón Ejecutar proyecto en la barra de herramientas de JBuilder para compilar y ejecutar la aplicación.

También puede seleccionar `HOLAATodosClase.java` en el panel del proyecto. Para ello, haga clic con el botón derecho del ratón y seleccione Ejecutar usando "HOLAATodosClase".

En primer lugar se abre el panel de mensajes, que muestra el proceso de ejecución. Es entonces cuando aparece la aplicación, que debe ofrecer el siguiente aspecto:



- Nota** La aplicación que se ejecuta en este tutorial refleja el aspecto Windows.
- 1 Seleccione Archivo | Salir en la aplicación “Hola a todos” para cerrarla.
 - 2 Pulse el botón Cerrar de la pestaña HolaATodosClase en el panel de mensajes para cerrar los mensajes.

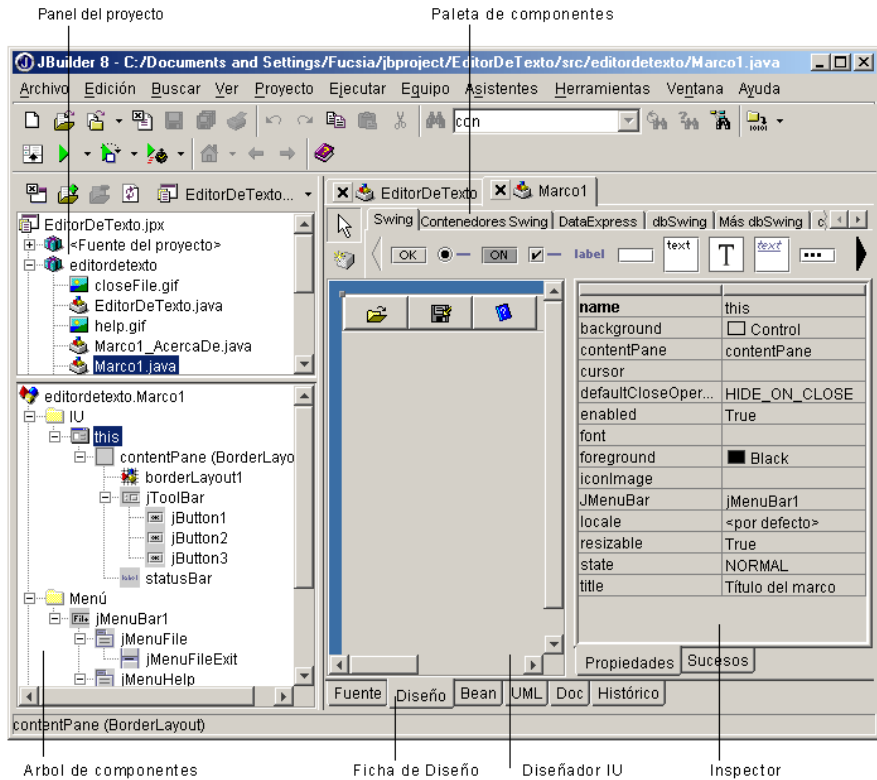
Paso 4: Personalización de la interfaz del usuario de la aplicación

Siga estos pasos para personalizar la interfaz de usuario de la aplicación:

- 1 Haga doble clic sobre `MarcoHolaATodos.java` en el panel del proyecto si todavía no está abierto.
- 2 Seleccione la pestaña Diseño de la parte inferior del panel de contenido para cambiar al modo de diseño.

El diseñador de interfaces de usuario aparece en el panel de contenido, con la paleta de componentes arriba y el Inspector a la derecha. La paleta de componentes se utiliza para añadir componentes a la interfaz, y el Inspector, para modificar propiedades y añadir sucesos al código. El panel de estructura, situado a la izquierda del panel de contenido, cuenta con un árbol de componentes con carpetas como IU, Menú y Otros.

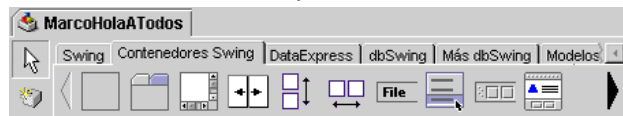
Figura 3.2 Elementos del diseñador de la interfaz de usuario



- 3 Haga clic en la pestaña Contenedores Swing de la paleta de componentes, encima del diseñador, y seleccione el componente JPanel para añadir un panel al diseño.

Sugerencia

Si se coloca el cursor sobre un componente de la paleta aparecen su nombre en un texto de ayuda inmediata.

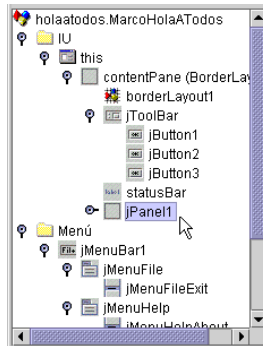


- 4 Coloque el componente en el centro del marco haciendo clic allí.

Nota

La propiedad `constraints` en el Inspector debe tener el valor `Center`. Si no es así, haga clic en la columna situada a la derecha de la propiedad `constraints` y seleccione `Center` en la lista desplegable.

jPanel1 se encuentra ahora seleccionado en el diseño de la aplicación y en el árbol de componentes.



Nota El componente seleccionado en el árbol de componentes o en el diseñador de la interfaz de usuario aparece en la barra de estado situada bajo el panel de estructura.

5 Asigne el color blanco al fondo de jPanel1.

- a Haga clic en la columna situada a la derecha de la propiedad `background` en el Inspector.
- b Haga clic en la *flecha abajo* para abrir la lista desplegable de colores y seleccione `White`, en la parte superior de la lista.

Nota Si pulsa demasiado a la derecha en la columna se abre el cuadro de diálogo Fondo. También se puede seleccionar el blanco introduciendo los números 255, que corresponden a los valores de rojo, verde y azul.

6 Añada un borde de línea a jPanel1 y cambie el color del borde a gris.

- a Haga clic en la columna situada a la derecha de la propiedad `border` en el Inspector.
- b Haga clic en la *flecha abajo* para abrir la lista desplegable de bordes y seleccione `Línea`.



- c Pulse el botón puntos suspensivos para acceder al cuadro de diálogo Borde.
- d Pulse `Black` en `Opciones | Color` para acceder a la lista desplegable de colores, y seleccione `Gray`.
- e Pulse `Aceptar` para cerrar el cuadro de diálogo Borde.

7 Cambie el administrador de diseño de jPanel1 a `null`.

- a Haga clic en la columna situada a la derecha de la propiedad `layout` en el Inspector.
- b Seleccione `null` en la lista desplegable.

La ausencia de diseño, o `null`, es la opción más adecuada para la creación de prototipos. Como el diseño `null` no utiliza un gestor de diseño, puede situar los componentes donde quiera. Sin embargo, dado que `null` utiliza la colocación absoluta de los componentes en lugar de la relativa, la interfaz de usuario no cambia de tamaño correctamente cuando se modifica el tamaño de la ventana de la aplicación. Por ello se recomienda no dejar nunca los contenedores en modo `null` para la distribución. Más adelante se explica la forma de cambiar a un diseño portable antes de distribuir la aplicación.

Nota

El panel de mensajes sólo aparece cuando se muestra un mensaje. Por lo tanto, aún no verá el panel de mensajes si está siguiendo estos pasos.

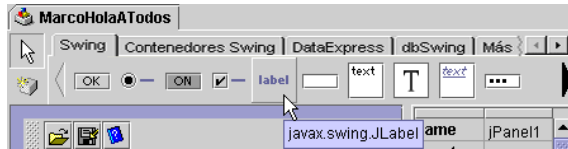


- 8 Seleccione Archivo | Guardar todo o pulse el icono Guardar todo en la barra de herramientas para guardar el proyecto.

Paso 5: Incorporación de componentes a la aplicación

Ahora utilice la paleta de componentes para añadir un componente `JLabel` al componente `JPanel`.

- 1 Pulse la pestaña Swing en la paleta de componentes y seleccione el componente `JLabel`.



- 2 Coloque el componente en el objeto `JPanel` en su diseño de una de las siguientes maneras:
 - En el árbol de componentes, pulse `jPanel1`. De esta forma se coloca en la esquina superior izquierda del panel.
 - Haga clic en el diseñador de interfaces de usuario, que contiene el componente `jPanel1`. La esquina superior izquierda del componente se coloca en el lugar en el que se hace clic.

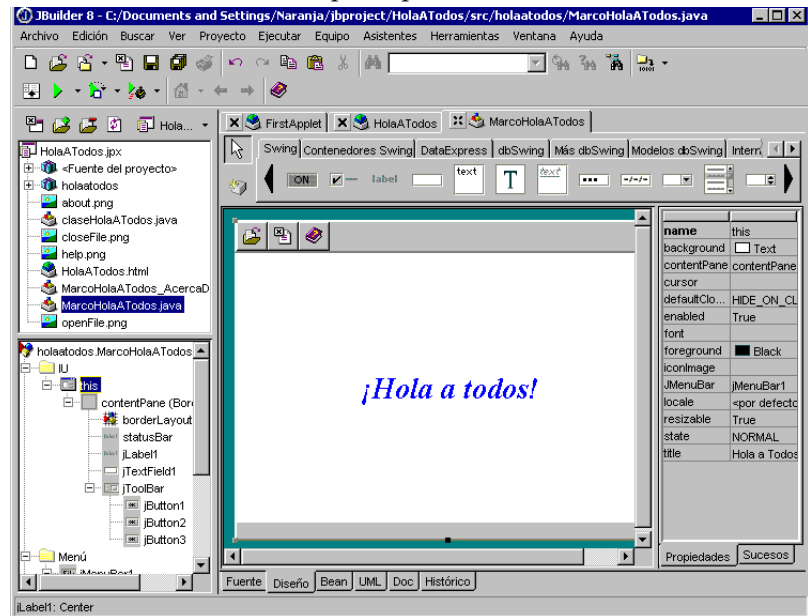
Observe que `jLabel1` se añade debajo de `jPanel1`, en el árbol de componentes. Si coloca el componente en un sitio erróneo, puede seleccionarlo en el árbol de componentes o en el diseñador y pulsar la tecla *Supr*. A continuación, vuelva a intentarlo.

- 3 En el diseñador, haga clic en el centro del componente de etiqueta y arrástrelo al centro del panel.
- 4 Seleccione `jLabel1` en el árbol de componentes y haga lo siguiente:
 - a Haga doble clic en la columna situada a la derecha de la propiedad `text` en el Inspector, escriba `jHola a todos!` y pulse *Intro*.

“¡Hola a todos!” aparece parcialmente en la etiqueta. No se preocupe si el texto no aparece completo en la etiqueta. Esto se puede resolver más adelante, cambiando la fuente.

- b Para seleccionar el tipo de letra, haga clic en la columna situada a la derecha de la propiedad `font`. Pulse el botón de puntos suspensivos para abrir el cuadro de diálogo Fuente.
- c Elija Serif en la lista de fuentes y marque Negrita y Cursiva. Escriba 28 en el cuadro Tamaño y pulse Aceptar.
- d Cambie el tamaño de `jLabel1` arrastrando los tiradores negros hasta que el texto “¡Hola a todos!” sea completamente visible.
- e Haga clic en la columna situada a la derecha de la propiedad `foreground` en el Inspector para definir el color del texto “¡Hola a todos!”. Pulse la *flecha abajo* y seleccione Blue en la lista desplegable de colores.

Ahora el diseño tiene un aspecto parecido a éste:



- 5 Seleccione Archivo | Guardar todo o pulse el icono Guardar todo en la barra de herramientas para guardar el proyecto.

Paso 6: Modificación del código fuente

Puede cambiar la información del cuadro Acerca de modificando directamente el código. La versión por defecto de la aplicación creada por el Asistente para aplicaciones es la 1.0.

- 1 Haga doble clic en `HolaATodosMarco_AcercaDe.java`, en el panel del proyecto, para abrir el archivo.

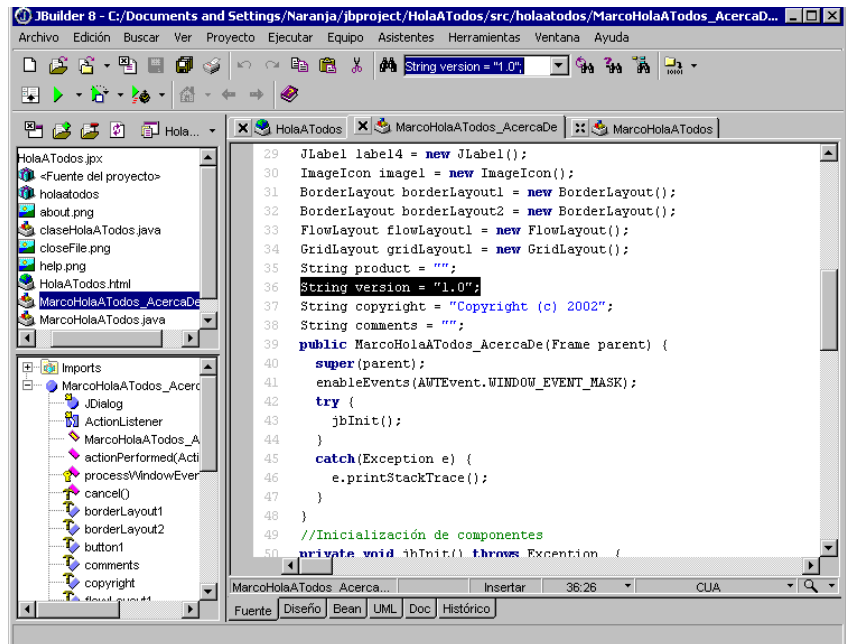
El panel de contenido cambia a la vista Fuente, donde se puede modificar el código con el editor.

- 2 Seleccione **Buscar | Buscar** para abrir el cuadro de diálogo **Buscar/Reemplazar texto**.
- 3 Introduzca la siguiente línea de código en el cuadro de diálogo **Buscar**:

```
String version = "1.0";
```

- 4 Pulse **Buscar**.

El editor encuentra el texto seleccionado.



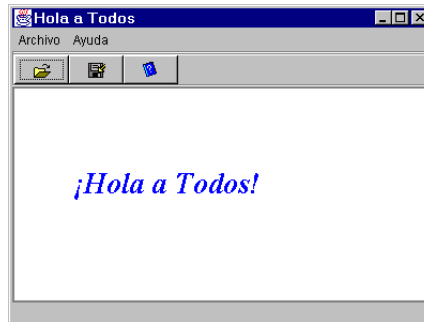
- 5 Seleccione `1.0` y escriba `2.0` dentro de las comillas.
- 6 Elija **Archivo | Guardar todo**.

Paso 7: Compilación y ejecución de la aplicación

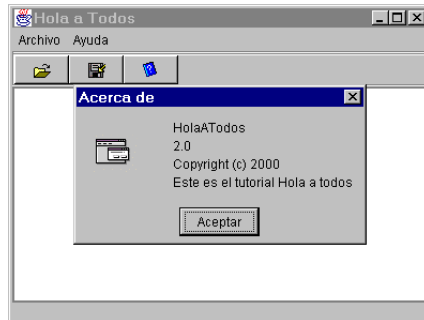
En este punto, la aplicación se puede compilar y ejecutar.

- 1 Seleccione Proyecto | Ejecutar Make del proyecto “HolaATodos.jpj” para compilar el proyecto.
- 2 Seleccione Ejecutar | Ejecutar proyecto.

Aparece la aplicación “Hola a todos”:



- 3 Elija Ayuda | Acerca de. El nuevo número de versión aparece en el cuadro de diálogo Acerca de.



- 4 Haga clic en Aceptar para cerrar el cuadro de diálogo.
- 5 Seleccione Archivo | Salir en la aplicación “Hola a todos” para cerrarla.

Paso 8: Ejecución de la aplicación desde la línea de comandos

También es posible ejecutar la aplicación desde fuera del entorno de JBuilder, por medio de la línea de comandos.

Nota El directorio `<jdk>/bin/`, que contiene el comando **java**, se debe utilizar en la vía de acceso. Si en la vía de acceso figura **java**, cuando se escribe `java` en la línea de comandos, debería aparecer información sobre el comando.

Si no se encuentra en la vía de acceso, es necesario ejecutar la aplicación desde el directorio `<jdk>/bin/`.

Para ejecutar la aplicación, siempre que **java** se encuentre en la vía de acceso:

- 1 Abra la ventana de línea de comandos.
- 2 Ejecute la aplicación con el siguiente comando en una sola línea:

```
java -classpath:  
/<inicio>/j bproject/HolaATodos/classes holaatodos.HolaATodosClase
```

Nota En Windows, se utiliza la barra invertida (\).

La estructura de este comando es la siguiente:

```
java -classpath nombre-de-paquete.nombre-de-clase-principal
```

En el ejemplo:

- `java` = ejecuta la aplicación en Java.
 - `-classpath` = una opción que establece la ruta de búsqueda de las clases y los recursos de la aplicación.
 - `classpath = /<inicio>/j bproject/HolaATodos/classes`
La vía de acceso a clases debe señalar al directorio que contiene las clases compiladas. En este ejemplo, el directorio `classes` se ha definido como vía de acceso a archivos generados de las clases compiladas en el primer paso del Asistente para proyectos.
 - `<inicio>` = el directorio inicial, como por ejemplo, `c:\winnt\profiles\
<nombre de usuario>`
 - `nombre-de-paquete` = `holaatodos`
 - `nombre-de-clase-principal` = `HolaATodosClase`
- 3 Cierre la aplicación “Hola a todos”.

Paso 9: Asignación de sucesos a botones

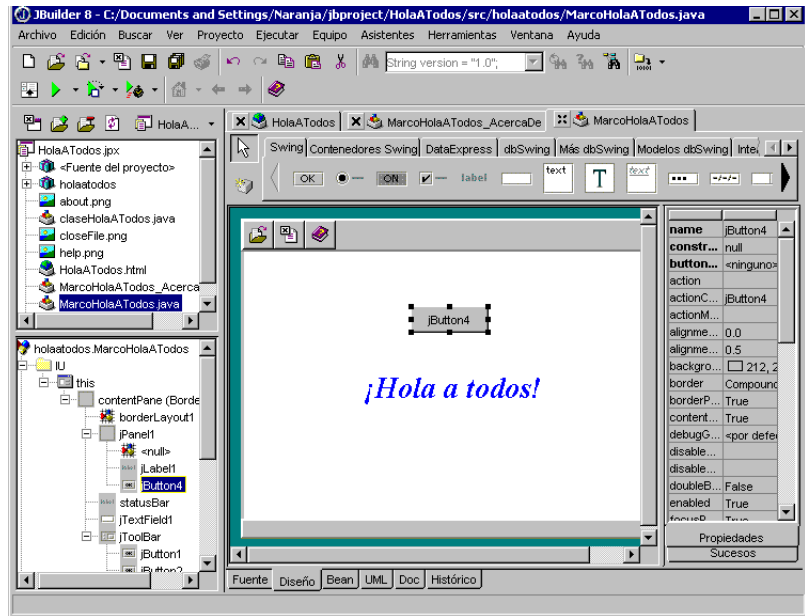
A continuación se va a añadir otro componente Swing a la aplicación.

- 1 Abra `HolaATodosMarco.java` y pulse la pestaña **Diseño** para pasar al diseñador de interfaces de usuario.



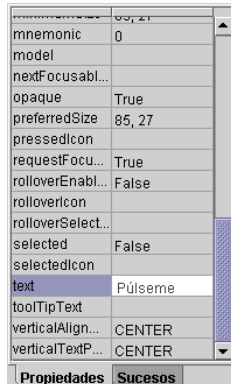
- 2 Haga clic en el componente `JButton`, en la pestaña **Swing** de la paleta de componentes, y suéltelo en `jPanel1`, en el árbol de componentes, o en el panel del diseño. `jButton4` se añade debajo de `jPanel1` en el árbol de componentes.

- Haga clic en JButton4 en el diseño y arrástrelo a la parte superior central del diseño, como se muestra en la siguiente imagen.



- En el Inspector, cambie la propiedad Text de JButton4 a Púlsame. Pulse Intro. Aumente el tamaño del botón arrastrando los tiradores negros hasta que la palabra “Púlsame” se vea correctamente.

El Inspector presentará el siguiente aspecto:



- Haga clic en la pestaña Sucesos del Inspector, para definir qué ocurrirá cuando se pulsa JButton4.

- 6 Haga doble clic en la columna que se encuentra a la derecha del suceso `ActionPerformed`.

JBuilder pasa al editor, donde se ha añadido el siguiente código de estructura correspondiente al suceso `ActionPerformed`

```
void jButton4_actionPerformed(ActionEvent e) {  
  
}
```

Ahora se puede introducir el código que define el suceso.

- 7 Escriba el siguiente código, marcado en negrita:

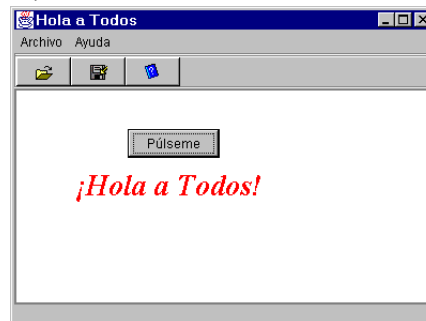
```
void jButton4_actionPerformed(ActionEvent e) {  
    jLabel1.setForeground(new Color(255,0,0));  
}
```

Sugerencia

Utilice CodeInsight para completar el código. Escriba `jLabel1.` y espere a que aparezca la ventana emergente, o pulse ***Ctrl+Espacio*** para abrirla. Si CodeInsight no aparece, compruebe las opciones de la ficha CodeInsight de Opciones del editor (Herramientas | Opciones del editor). Escriba `setfor` para resaltar `setForeground(Color)` en la ventana emergente, o emplee las flechas. Pulse ***Intro***.

Ahora, cuando ejecute la aplicación y pulse el botón “Pulseme”, la frase “¡Hola a todos!” aparecerá en rojo.

- 8 Elija Archivo | Guardar todo.
- 9 Seleccione Proyecto | Ejecutar Make del proyecto “HolaATodos.jpx”.
- 10 Seleccione Ejecutar | Ejecutar proyecto.
- 11 Pulse el botón “Pulseme”. El color del texto “¡Hola a todos!” cambia a rojo.



- 12 Seleccione Archivo | Salir para cerrar la aplicación “¡Hola a todos!”.

Paso 10: Cómo completar la interfaz de usuario

El último paso para completar la aplicación consiste en aplicarle un diseño portable. Recuerde que si se deja `jPanel1` en el valor `null`, esto es, sin diseño, los componentes de la interfaz no se colocan correctamente cuando el usuario cambia el tamaño de la ventana de la aplicación mientras se ejecuta el programa. Dado que `null` utiliza las posiciones absolutas, los componentes permanecen en el mismo sitio independientemente del tamaño de la ventana. Por ello, `null` no es un gestor de diseño adecuado para la distribución final de la aplicación.

Para comprobar los problemas que ocasiona el uso de `null`, ejecute la aplicación como sigue:

- 1 Haga clic con el botón derecho del ratón en `HolaATodosClase.java`, que contiene el método `main()`, y elija Ejecutar.
- 2 Cambie el tamaño de la ventana de la aplicación “Hola a todos” y observe el comportamiento de los componentes.

Observe que los componentes de las etiquetas y botones no se desplazan cuando se redimensiona la ventana, ya que su posición es absoluta y no relativa con respecto al tamaño de la ventana.



- 3 Cierre la aplicación “Hola a todos”.

Los diseños portables, a diferencia de `null`, utilizan las posiciones relativas, que son dinámicas. Cuando se emplea un diseño portable, los componentes cambian de posición correctamente cuando el usuario cambia el tamaño de la ventana de la aplicación. En este ejemplo se va a cambiar el diseño a `GridBagLayout`.

Si desea modificar el comportamiento del diseño:

- 1 Vuelva al archivo `HolaATodosMarco.java` del panel de contenido y haga doble clic en la pestaña **Diseño** para pasar al diseñador.
 - 2 En el árbol de componentes, escoja `jPanel1`.
 - 3 Abra la pestaña **Propiedades del Inspector**.
 - 4 Cambie el diseño de `jPanel1` a `GridBagLayout` en el Inspector. Seleccione los distintos componentes en el diseñador y observe la zona que ocupan en la cuadrícula.
- `GridBagLayout` es una opción adecuada para colocar en una cuadrícula componentes de distintos tamaños.
- 5 Guarde la aplicación y ejecútela.
 - 6 Cambie el tamaño de la ventana de la aplicación y observe cómo cambian de tamaño los componentes.



- 7 Cierre la aplicación “Hola a todos”.

En los pasos 11 y 12 se emplean funciones de JBuilder SE y Enterprise. Si está utilizando JBuilder Personal, ya ha completado el tutorial.

Paso 11: Preparación de la aplicación para la distribución

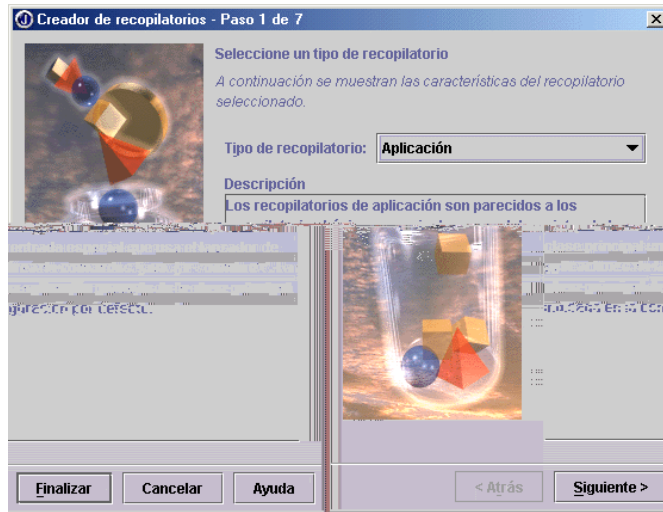
Continúe con el tutorial si
dispone de JBuilder SE o
Enterprise.

El Creador de recopilatorios reúne todos los archivos necesarios para distribuir el applet y los comprime en un archivo JAR (Java ARchive).

Para distribuir la aplicación:

- 1 Elija **Asistentes | Creador de recopilatorios** para abrir el Creador de recopilatorios.

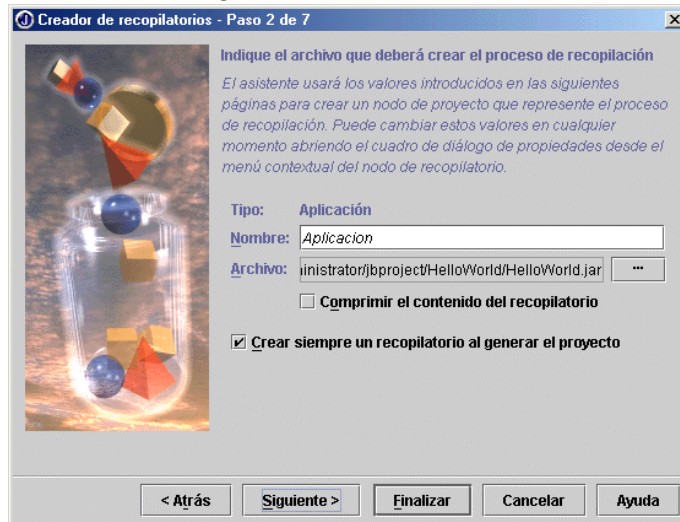
2 Seleccione Aplicación de la lista desplegable Tipo de archivo.



3 Pulse Siguiente para pasar al Paso 2.

4 Acepte las opciones por defecto del Paso 2.

Hol aTodos.jar se guardará en el directorio Hol aTodos.



5 Acepte las opciones por defecto de los Pasos 3 al 6, y pulse Siguiente en todos ellos.

6 Pulse Finalizar en el Paso 7 para cerrar el Creador de recopilatorios.

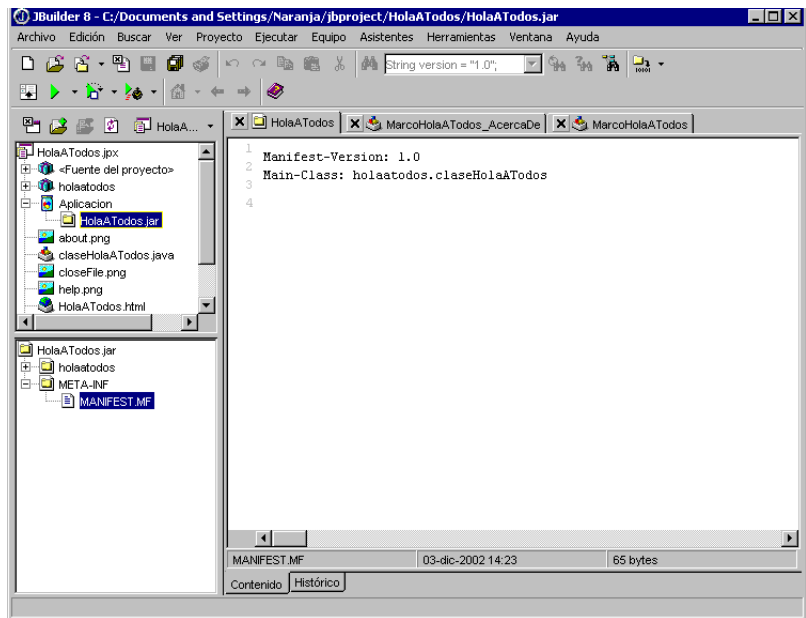
En el panel del proyecto aparece un nodo de recopilatorio denominado Apl i cación. Para modificar este archivo haga clic en él con el botón derecho y seleccione Propiedades.

- 7 Para compilar la aplicación y crear el archivo JAR haga clic con el botón derecho del ratón el nodo Aplicación y elija Ejecutar Make, o seleccione Proyecto | Ejecutar Make del proyecto “HolaATodos.jpj”.

El Creador de recopilatorios reúne todos los archivos en la vía de salida del proyecto (Proyecto | Propiedades de proyecto | Vías de acceso) dentro del archivo JAR.

- 8 Pulse el icono de ampliación situado junto al nodo del recopilatorio Aplicación para ampliarlo y ver el archivo HolaATodos.jar.
- 9 Haga doble clic en el archivo JAR, en el panel del proyecto.

El archivo descriptor aparece en el panel de contenido y el contenido del archivo JAR aparecen en el panel de estructura.



- 10 Guarde el proyecto.

Para obtener más información sobre los paquetes, consulte “Distribución de programas en Java” en *Creación de aplicaciones con JBuilder*.

Paso 12: Ejecución de la aplicación distribuida desde la línea de comandos

Si desea comprobar la aplicación distribuida, puede ejecutar el archivo JAR desde la línea de comandos.

Nota El directorio `<jdk>/bin/`, que contiene el comando **java**, se debe utilizar en la vía de acceso. Si en la vía de acceso figura **java**, cuando se escribe `java` en la línea de comandos, debería aparecer información sobre el comando. Si no se encuentra en la vía de acceso, es necesario ejecutar la aplicación desde el directorio `<jdk>/bin/`.

- 1 Abra la ventana de línea de comandos.
- 2 Escriba el siguiente comando en una sola línea del indicador de comandos:

```
java -classpath /<inicio> /jbproject/HolaATodos/HolaATodos.jar  
holaatodos.HolaATodosClase
```

Nota En Windows, se utiliza la barra invertida (`\`).

El comando debe tener la siguiente estructura:

```
java -classpath nombre-de-paquete.nombre-de-clase-principal
```

En el ejemplo:

- `java` = ejecuta la aplicación en Java.
- `-classpath` = una opción que establece la ruta de búsqueda de las clases y los recursos de la aplicación.
- `classpath = /<raíz> /jbproject/HolaATodos/HolaATodos.jar`
La vía de acceso a clases debe incluir el archivo JAR y su posición correcta. En este ejemplo, el archivo JAR se encuentra en el directorio del proyecto, `HolaATodos`. Por defecto, el Creador de recopilatorios lo guarda en este directorio.
- `<raíz>` = el directorio inicial, como por ejemplo, `c:\winnt\profiles\
<nombre de usuario>`
- `nombre-de-paquete` = `holaatodos`
- `nombre-de-clase-principal` = `HolaATodosClase`

Si desea obtener más información sobre los archivos JAR, consulte el tutorial de JAR en

<http://java.sun.com/docs/books/tutorial/jar/index.html>.

- 3 La aplicación “Hola a todos” se carga y se ejecuta.

¡Enhorabuena, ha creado su primera aplicación con JBuilder! Ahora que ya se ha familiarizado con el entorno de desarrollo de JBuilder, descubrirá que sus múltiples funciones le permiten ahorrar tiempo y programar con mayor comodidad.

Si desea seguir más tutoriales relacionados con el diseño de interfaces en JBuilder, consulte “Editor de texto”, “GridBagLayout” y “Diseños anidados” en *Diseño de aplicaciones con JBuilder*.

Código fuente de HolaATodos

Código fuente de HolaATodosMarco.java

```
package holaatodos;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border*;

/**
 * Título:      HolaATodos
 * Descripción: Éste es el tutorial "Hola a todos".
 * Copyright:   Copyright (c) 2001
 * Empresa:    Mi Empresa
 * @author     Mi Nombre
 * @versión    1.0
 */

public class HelloWorldFrame extends JFrame {
    JPanel contentPane;
    JMenuBar jMenuBar1 = new JMenuBar();
    JMenu jMenuFile = new JMenu();
    JMenuItem jMenuFileExit = new JMenuItem();
    JMenu jMenuHelp = new JMenu();
    JMenuItem jMenuHelpAbout = new JMenuItem();
    JToolBar jToolBar = new JToolBar();
    JButton jButton1 = new JButton();
    JButton jButton2 = new JButton();
    JButton jButton3 = new JButton();
    ImageIcon image1;
    ImageIcon image2;
    ImageIcon image3;
    JLabel statusBar = new JLabel();
    BorderLayout borderLayout1 = new BorderLayout();
    JPanel jPanel1 = new JPanel();
    Border border1;
    JLabel jLabel1 = new JLabel();
    JButton jButton4 = new JButton();
    GridBagLayout gridBagLayout1 = new GridBagLayout();
}
```

```

/**Generar el marco*/
public HelloWorldFrame() {
    enableEvents(AWTEvent.WINDOW_EVENT_MASK);
    try {
        jbInit();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}

/**Iniciación de componente*/
private void jbInit() throws Exception{
    image1 = new ImageIcon
(holaatodos. HolaATodosMarco.class.getResource("openFile.gif"));
    image2 = new ImageIcon
(holaatodos. HolaATodosMarco.class.getResource("closeFile.gif"));
    image3 = new ImageIcon
(holaatodos. HolaATodosMarco.class.getResource("help.gif"));

    //setIconImage(Toolkit.getDefaultToolkit().createImage
(HolaATodosMarco.class.getResource("[El icono]")));
    contentPane = (JPanel) this.getContentPane();
    border1 = BorderFactory.createLineBorder(Color.gray, 2);
    contentPane.setLayout(borderLayout1);
    this.setSize(new Dimension(400, 300));
    this.setTitle("Titulo del marco");
    statusBar.setText(" ");
    jMenuFile.setText("File");
    jMenuFileExit.setText("Exit");
    jMenuFileExit.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            jMenuFileExit_actionPerformed(e);
        }
    });
    jMenuHelp.setText("Ayuda");
    jMenuHelpAbout.setText("Acerca de");
    jMenuHelpAbout.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            jMenuHelpAbout_actionPerformed(e);
        }
    });
    jButton1.setIcon(image1);
    jButton1.setToolTipText("Abrir archivo");
    jButton2.setIcon(image2);
    jButton2.setToolTipText("Cerrar archivo");
    jButton3.setIcon(image3);
    jButton3.setToolTipText("Ayuda");
    jPanel1.setBackground(Color.white);
    jPanel1.setBorder(border1);
    jPanel1.setLayout(gridBagLayout1);
    jLabel1.setFont(new java.awt.Font("Serif", 3, 28));
    jLabel1.setForeground(Color.blue);
    jLabel1.setText("¡Hola a todos!");
    jButton4.setText("Púlsame");

```

```

jButton4.addActionListener(new java.awt.event.ActionListener() {

    public void actionPerformed(ActionEvent e) {
        jButton4_actionPerformed(e);
    }
});
jToolBar.add(jButton1);
jToolBar.add(jButton2);
jToolBar.add(jButton3);
jMenuFile.add(jMenuFileExit);

jMenuHelp.add(jMenuHelpAbout);
jMenuBar1.add(jMenuFile);
jMenuBar1.add(jMenuHelp);
this.setJMenuBar(jMenuBar1);
contentPane.add(jToolBar, BorderLayout.NORTH);
contentPane.add(statusBar, BorderLayout.SOUTH);
contentPane.add(jPanel1, BorderLayout.CENTER);
jPanel1.add(jLabel1, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.0
    ,GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(45, 125,
    102, 110), 10, -6));
jPanel1.add(jButton4, new GridBagConstraints(0, 0, 1, 1, 0.0, 0.0
    ,GridBagConstraints.CENTER, GridBagConstraints.NONE, new Insets(46, 148,
    0, 149), 16, 0));
}

/**Acción de Archivo | Salir realizada*/
public void jMenuFileExit_actionPerformed(ActionEvent e) {
    System.exit(0);
}

/**Realizar Ayuda|Acerca de*/
public void jMenuHelpAbout_actionPerformed(ActionEvent e) {
    HelloWorldFrame_AboutBox dlg = new HelloWorldFrame_AboutBox(this);
    Dimension dlgSize = dlg.getPreferredSize();
    Dimension frmSize = getSize();
    Point loc = getLocation();
    dlg.setLocation((frmSize.width - dlgSize.width) / 2 + loc.x, (frmSize.height -
        dlgSize.height) / 2 + loc.y);
    dlg.setModal(true);
    dlg.show();
}

/**Modificado de forma que se pueda salir cuando se cierre la ventana*/
protected void processWindowEvent(WindowEvent e) {
    super.processWindowEvent(e);
    if(e.getID() == WindowEvent.WINDOW_CLOSING) {
        jMenuFileExit_actionPerformed(null);
    }
}

void jButton4_actionPerformed(ActionEvent e) {
    jLabel1.setForeground(new Color(255, 0, 0));
}
}

```

Código fuente de ClaseHolaATodos.java

```

package holaatodos;

import javax.swing.UIManager;
import java.awt.*;

/**
 * Título:      HolaATodos
 * Descripción: Éste es el tutorial "Hola a todos".
 * Copyright:   Copyright (c) 2001
 * Empresa:     Mi Empresa
 * @author      Mi Nombre
 * @versión 1.0
 */

public class HelloWorldClass {
    boolean packFrame = false;

    /**Crear la aplicación*/
    public HelloWorldClass() {
        HelloWorldFrame frame = new HelloWorldFrame();
        //Validar marcos que tienen tamaños preestablecidos
        //Empaquetar marcos con información de tamaño preferente útil, p. ej. del
        diseño
        if (packFrame) {
            frame.pack();
        }
        else {
            frame.validate();
        }
        //Centrar la ventana
        Dimension screenSize = Toolkit.getDefaultToolkit().getScreenSize();
        Dimension frameSize = frame.getSize();
        if (frameSize.height > screenSize.height) {
            frameSize.height = screenSize.height;
        }
        if (frameSize.width > screenSize.width) {
            frameSize.width = screenSize.width;
        }
        frame.setLocation((screenSize.width - frameSize.width) / 2,
            (screenSize.height - frameSize.height) / 2);
        frame.setVisible(true);
    }
    /**método Main*/
    public static void main(String[] args) {
        try {
            UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        new HelloWorldClass();
    }
}

```

Código fuente de HolaATodosMarco_AboutBox.java

```

package holaatodos;

import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.border*;

/**
 * Título:      HolaATodos
 * Descripción: Este es el tutorial "Hola a todos".
 * Copyright:   Copyright (c) 2001
 * Empresa:    Mi Empresa
 * @author     Mi Nombre
 * @versión 1.0
 */

public class HelloWorldFrame_AboutBox extends JDialog implements ActionListener {

    JPanel panel1 = new JPanel();
    JPanel panel2 = new JPanel();
    JPanel insetsPanel1 = new JPanel();
    JPanel insetsPanel2 = new JPanel();
    JPanel insetsPanel3 = new JPanel();
    JButton button1 = new JButton();
    JLabel imageLabel = new JLabel();
    JLabel label1 = new JLabel();
    JLabel label2 = new JLabel();
    JLabel label3 = new JLabel();
    JLabel label4 = new JLabel();
    BorderLayout borderLayout1 = new BorderLayout();
    BorderLayout borderLayout2 = new BorderLayout();
    FlowLayout flowLayout1 = new FlowLayout();
    GridLayout gridLayout1 = new GridLayout();
    String product = "";
    String version = "2.0";
    String copyright = "Copyright (c) 2001";
    String comments = "";
    public HelloWorldFrame_AboutBox(Frame parent) {
        super(parent);
        enableEvents(AWTEvent.WINDOW_EVENT_MASK);
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
        pack();
    }
    /**Iniciación de componente*/
    private void jbInit() throws Exception{

```

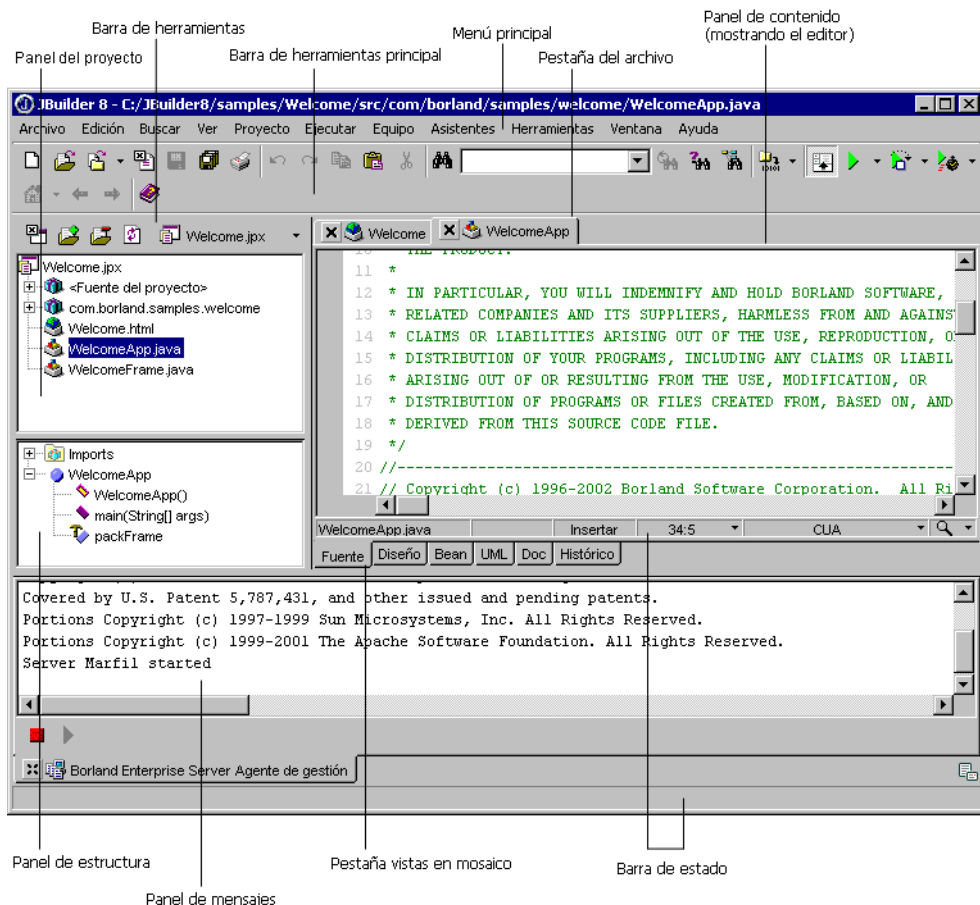
```

        //imageLabel.setIcon
        //(new ImageIcon(HelloWorldFrame_AboutBox.class.getResource
        //("El icono"))));
        this.setTitle("Acerca de");
        setResizable(false);
        panel1.setLayout(borderLayout1);
        panel2.setLayout(borderLayout2);
        insetsPanel1.setLayout(flowLayout1);
        insetsPanel2.setLayout(flowLayout1);
        insetsPanel2.setBorder(BorderFactory.createEmptyBorder(10, 10, 10, 10));
        gridLayout1.setRows(4);
        gridLayout1.setColumns(1);
        label1.setText(product);
        label2.setText(version);
        label3.setText(copyright);
        label4.setText(comments);
        insetsPanel3.setLayout(gridLayout1);
        insetsPanel3.setBorder(BorderFactory.createEmptyBorder(10, 60, 10, 10));
        button1.setText("Aceptar");
        button1.addActionListener(this);
        insetsPanel2.add(imageLabel, null);
        panel2.add(insetsPanel2, BorderLayout.WEST);
        this.getContentPane().add(panel1, null);
        insetsPanel3.add(label1, null);
        insetsPanel3.add(label2, null);
        insetsPanel3.add(label3, null);
        insetsPanel3.add(label4, null);
        panel2.add(insetsPanel3, BorderLayout.CENTER);
        insetsPanel1.add(button1, null);
        panel1.add(insetsPanel1, BorderLayout.SOUTH);
        panel1.add(panel2, BorderLayout.NORTH);
    }
    /**Modificado de forma que se pueda salir cuando se cierre la ventana*/
    protected void processWindowEvent(WindowEvent e) {
        if(e.getID() == WindowEvent.WINDOW_CLOSING) {
            cancel();
        }
        super.processWindowEvent(e);
    }
    /**Cierre el cuadro de diálogo*/
    void cancel() {
        dispose();
    }
    /**Cierra el diálogo cuando hay un suceso botón*/
    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == button1) {
            cancel();
        }
    }
}

```

El Visualizador de aplicaciones

JBuilder sólo utiliza una ventana para realizar la mayoría de las funciones de desarrollo: modificación, diseño visual, desplazamiento, búsqueda, compilación, depuración, etc. Esta ventana se denomina Visualizador de aplicaciones y contiene varios paneles en los que se efectúan estas funciones de desarrollo. Según el tipo de archivo seleccionado en el panel del proyecto se muestran unos u otros paneles con pestañas en la parte inferior del panel de contenido.

Figura 4.1 La ventana Visualizador de aplicaciones

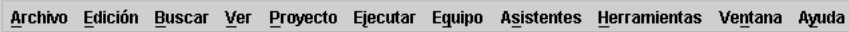
Puede abrir varios visualizadores de aplicaciones a través de Ventana | Nueva ventana. En JBuilder SE y Enterprise también es posible pulsar un archivo del panel del proyecto con el botón derecho del ratón y seleccionar Abrir en nueva ventana.

Todas las instancias del Visualizador de aplicaciones muestran los mismos proyectos abiertos, pero se pueden abrir archivos distintos en los paneles de contenido. Todas las instancias del Visualizador de aplicaciones están sincronizadas. Esto significa que si, por ejemplo, el mismo archivo está abierto en dos instancias, los cambios efectuados en una de ellas se reflejan en la otra.

Utilice el menú Ver para decidir qué paneles desea que se muestren en el Visualizador de aplicaciones actual.

Menú principal

El menú principal se encuentra en la parte superior del Visualizador de aplicaciones. El menú Equipo es una función de JBuilder SE y Enterprise. Si desea información sobre los distintos menús, consulte “Menús de JBuilder”.



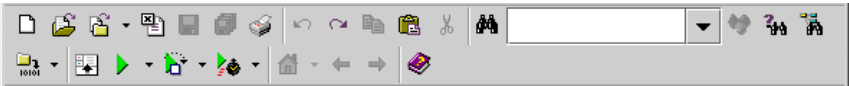
Barra de herramientas

La barra de herramientas principal se encuentra en la parte superior del Visualizador de aplicaciones, bajo la barra de menús. Se compone de barras de herramientas menores agrupadas por funciones: Archivo, Edición, Buscar, Generar, Ejecutar/Depurar, Desplazamiento y Ayuda. El contenido de la barra de herramientas se puede modificar activando y desactivando diversas opciones del menú Ver | Barras de herramientas.

Las funciones disponibles varían ligeramente según la edición de JBuilder

Si desea averiguar qué funciones se encuentran disponibles en la barra de herramientas de su edición de JBuilder, consulte “Menús de JBuilder”.

La barra de herramientas que se muestra aquí aparece plegada en dos líneas; es probable que en su caso ofrezca un aspecto ligeramente distinto, según la anchura de la ventana del Explorador de aplicaciones.



Las barras de herramientas cuentan con botones de método abreviado para los siguientes comandos de menú:

Tabla 4.1 Botones de la barra de herramientas

Icono	Menú equivalente	Descripción
	Archivo Nuevo	Abre la galería de objetos, en la que se puede elegir entre distintos asistentes.
	Archivo Abrir	Abre un proyecto, un archivo o un paquete.
	Archivo Abrir de nuevo	Vuelve a abrir un proyecto, un archivo o un paquete. Permite seleccionar un elemento de una lista.
	Archivo Cerrar	Cierra el archivo actual.
	Archivo Guardar archivo	Guarda el archivo actual.
	Archivo Guardar todo	Guarda todos los proyectos y archivos abiertos, conservando sus nombres.

Tabla 4.1 Botones de la barra de herramientas (continuación)




















Icono	Menú equivalente	Descripción
	Archivo Imprimir	Imprime el archivo o el texto seleccionado.
	Edición Deshacer	En el editor, reescribe los caracteres borrados, borra los caracteres insertados, recupera los caracteres sobrescritos y devuelve el cursor a la posición anterior. Deshace las acciones efectuadas en los diseñadores. La acción Deshacer cuenta con muchos niveles.
	Edición Volver a hacer	Invierte las operaciones realizadas por Deshacer. Existen muchos niveles de la acción Volver a hacer.
	Edición Copiar	Copia en el portapapeles el texto seleccionado en el editor o el objeto seleccionado en el diseñador de interfaces.
	Edición Pegar	Pega el contenido del portapapeles en la posición del cursor.
	Edición Cortar	Corta en el portapapeles el texto seleccionado en el editor o el objeto seleccionado en el diseñador de interfaces.
	Buscar Buscar	Busca texto en el archivo actual. Escriba todo en minúscula si no desea que la búsqueda distinga entre mayúsculas y minúsculas, o utilice mayúsculas para que se realice la distinción. Las búsquedas anteriores se pueden seleccionar en el menú desplegable.
	Buscar Buscar de nuevo	Busca la siguiente aparición de una cadena en el archivo actual.
	Buscar Reemplazar	Sustituye un texto por otro en el archivo actual.
	Buscar Buscar clases	Carga en el Visualizador de aplicaciones la clase especificada. La clase debe encontrarse en la vía de acceso de importación del archivo actual.
	Proyecto Ejecutar Make	Compila los recursos y los archivos .java del proyecto seleccionado, si los archivos .class correspondientes no existen o han quedado obsoletos. También compila los archivos importados de los que depende el proyecto, si sus archivos .class no existen o han quedado obsoletos. La flecha abre un menú desplegable que permite seleccionar Ejecutar Make (la opción por defecto), Generar de nuevo o un objetivo de creación definido por el usuario.
	Ver Mensajes	Conmuta la presentación del panel de mensajes.

Tabla 4.1 Botones de la barra de herramientas (continuación)

Icono	Menú equivalente	Descripción
	Ejecutar Ejecutar proyecto	Compila y ejecuta la aplicación con la configuración por defecto especificada en la ficha Ejecutar del cuadro de diálogo Propiedades de proyecto. Pulse la flecha para seleccionar una configuración distinta en la lista desplegable.
	Ejecutar Depurar proyecto	Depura el programa con la configuración de ejecución por defecto seleccionada en la ficha Ejecutar del cuadro de diálogo Propiedades de proyecto. Pulse Nuevo o Edición si desea añadir o modificar una configuración.
	Ejecutar Optimizar proyecto	Evalúa el código del proyecto mediante Optimizeit. Esta opción sólo se encuentra disponible si se tiene Optimizeit instalada.
	Ver Histórico	Vuelve a la clase o la ficha inicial de la lista. Pulse el botón de flecha, a la derecha del botón, para acceder a la lista de historial.
	Ver Atrás	Desplaza el visualizador al elemento anterior de la lista de historial.
	Ver Adelante	Desplaza el visualizador al elemento siguiente de la lista de historial.
	Ayuda Temas de ayuda	Abre el visualizador de ayuda.

Panel del proyecto

Para desarrollar programas en el entorno de JBuilder se debe crear un proyecto. Los proyectos de JBuilder organizan los archivos utilizados y mantienen las propiedades definidas. JBuilder guarda la configuración del proyecto en un *archivo del proyecto*, un archivo de tipo XML con la extensión .jpx. El archivo de proyecto se modifica automáticamente con los cambios que se realizan; el usuario no debe acceder a él directamente. El archivo del proyecto activo es el nodo superior del panel del proyecto.

El panel del proyecto muestra el contenido del proyecto activo. Consta de los siguientes elementos:

- Una pequeña barra de herramientas con cuatro botones:
 - Cerrar proyecto: cierra el proyecto activo.
 - Añadir archivos/paquetes: abre el cuadro de diálogo Añadir archivos o paquetes al proyecto.

- Eliminar del proyecto: elimina del proyecto los archivos seleccionados.
- Actualizar: actualiza el proyecto y busca automáticamente paquetes de código (Proyecto | Propiedades de proyecto | General).
- Una lista desplegable de todos los proyectos abiertos.
- Una vista en árbol con el contenido del proyecto activo.
- Un menú emergente sensible al contexto para cada *nodo* (archivo, paquete o directorio) del panel del proyecto, al que se puede acceder pulsando el elemento con el botón derecho del ratón.

Creación de proyectos

Si desea crear un proyecto, seleccione Archivo | Nuevo proyecto en el menú principal, o elija Archivo | Nuevo, pulse el icono Proyecto de la ficha Proyecto de la galería de objetos y, por último, haga doble clic en el icono Proyecto. Si desea crear un proyecto a partir de código y recursos ya creados, pulse el icono Proyecto para el código existente de la misma ficha.

Adición de archivos y clases a un proyecto

Si desea añadir al proyecto archivos o clases ya creados, pulse el botón Añadir Archivos/Paquetes de la barra de herramientas del panel del proyecto y busque el archivo o el paquete que desee.

Para crear un archivo de clase de código fuente de Java, elija Archivo | Nueva clase para abrir el Asistente para clases, con la clase creada en el proyecto actual.

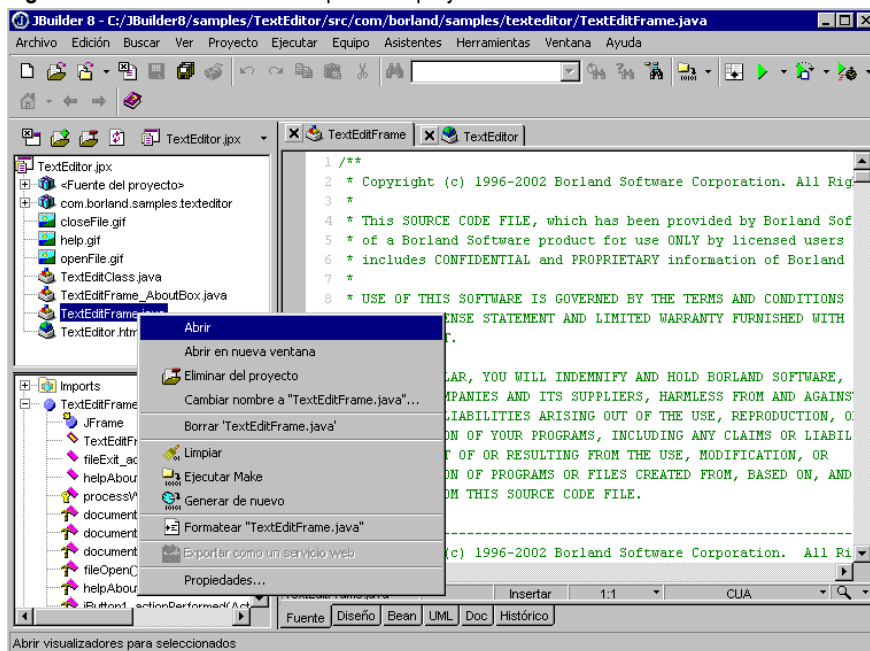
Si desea añadir un archivo vacío al proyecto, seleccione Archivo | Archivo nuevo para abrir el cuadro de diálogo Crear archivo, escriba el nombre del archivo en el campo Nombre y seleccione su tipo en la lista desplegable Tipo. También puede utilizar el campo Directorio para cambiar la ubicación donde se guarda el archivo. Cuando se pulsa el botón Aceptar, el archivo vacío se añade al proyecto.

Si se utiliza el Asistente para aplicaciones, que se encuentra en la ficha General de la galería de objetos, la nueva aplicación se añade al proyecto actual.

El panel del proyecto

El árbol del proyecto se puede recorrer libremente. Aquí se pueden efectuar selecciones múltiples de archivos, así como hacer clic con el botón derecho del ratón y elegir elementos del menú contextual, sin necesidad de abrir ningún archivo. Las opciones del menú contextual varían según el tipo del archivo seleccionado. Para abrir un archivo en el panel de contenido del Visualizador de aplicaciones, selecciónelo en el panel del proyecto y pulse *Intro* o haga doble clic en él.

Figura 4.2 Menú contextual del panel del proyecto



La opciones que aparecen en el menú del panel del proyecto dependen del contexto y de la versión de JBuilder que se esté utilizando.

En el panel del proyecto se pueden abrir todos los proyectos deseados, pero sólo uno es el activo en cada momento. El proyecto activo se muestra seleccionado en la lista desplegable del panel del proyecto.

Sugerencia

Para buscar rápidamente un archivo o un paquete en el panel del proyecto, pase el foco al árbol del proyecto y empiece a escribir.

Consulte

- [Capítulo 7, “Proyectos de JBuilder”](#) y “Creación y gestión de proyectos” en *Creación de aplicaciones con JBuilder* si desea información adicional sobre los proyectos.

- [“Búsqueda en árboles” en la página 4-18](#) si desea más información sobre la búsqueda en el árbol del proyecto.
- “Filtrado de paquetes” en *Creación de aplicaciones con JBuilder* si desea instrucciones para filtrar paquetes de modo que sólo se muestren los deseados en el panel del proyecto.

Si desea más información sobre la forma de trabajar con proyectos, consulte “Creación y gestión de proyectos” en *Creación de aplicaciones con JBuilder*.

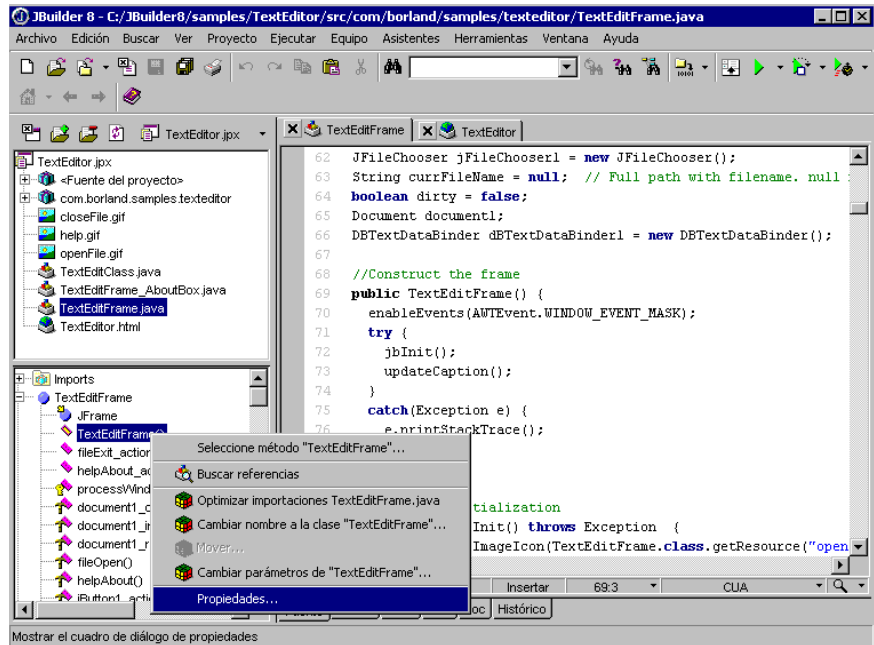
El panel de estructura

El panel de estructura muestra la estructura del archivo que se encuentra seleccionado en el panel de contenido. Esta estructura se presenta en forma de árbol, con todos los miembros y campos del archivo. Cuando así se requiere, el panel de estructura muestra una carpeta **Errores** que contiene todos los errores de sintaxis, una carpeta **Importaciones** que contiene una lista de los paquetes importados y una carpeta **Por hacer** que contiene los comentarios de la etiqueta `@todo` de Javadoc.

Para buscar rápidamente un elemento de un archivo, pase el foco al árbol del panel de estructura y empiece a escribir el nombre del elemento deseado. Para obtener más información, consulte [“Búsqueda en árboles” en la página 4-18](#).

Ordenación en el panel de estructura

Puede cambiar el orden de clasificación del panel de estructura y el retardo en el análisis en el cuadro de diálogo **Propiedades de visualización de la estructura**. Haga clic con el botón derecho del ratón en el panel de estructura y elija **Propiedades** para abrir este cuadro de diálogo y modificar las opciones.

Figura 4.3 Propiedades de visualización de la estructura

Estas opciones también están disponibles en la ficha Estructura Java del cuadro de diálogo Opciones del editor (Herramientas | Opciones del editor). Si lo prefiere, pulse el botón Ayuda del cuadro de diálogo Opciones del editor.



Presentación de la información de Javadoc

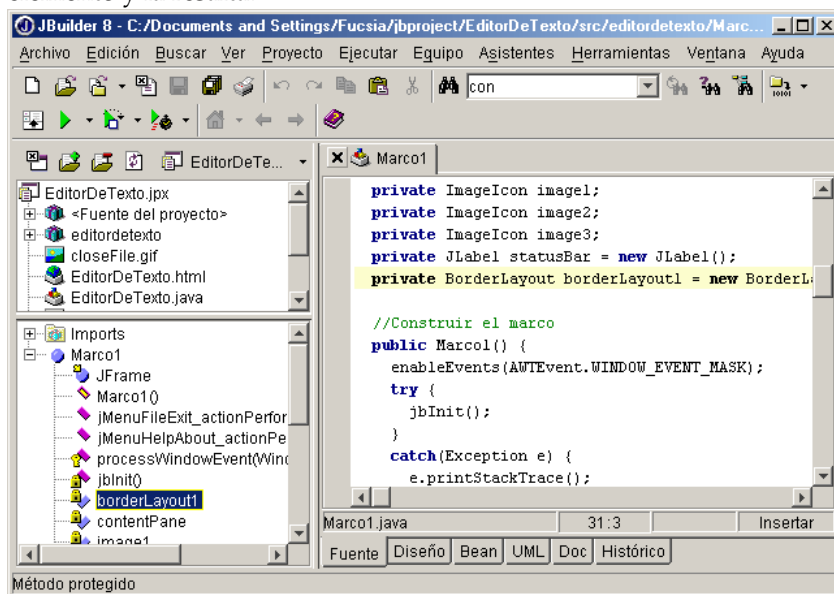
Las etiquetas `@todo` de los comentarios Javadoc también aparecen en el panel de estructura, en la carpeta `Por hacer`. Los conflictos de Javadoc aparecen en la carpeta `Conflictos Javadoc`.

Consulte

- “Utilización de etiquetas `@todo` en el editor” en la página 5-11
- “Conflictos en comentarios Javadoc” en *Creación de aplicaciones con JBuilder*

Desplazamiento por el código fuente

Además de presentar la estructura de la clase, el panel de estructura permite desplazarse rápidamente a clases, métodos y miembros del código fuente. Cuando se selecciona un elemento en el panel de estructura, el panel de contenido se desplaza hasta la línea que define el elemento y la resalta.



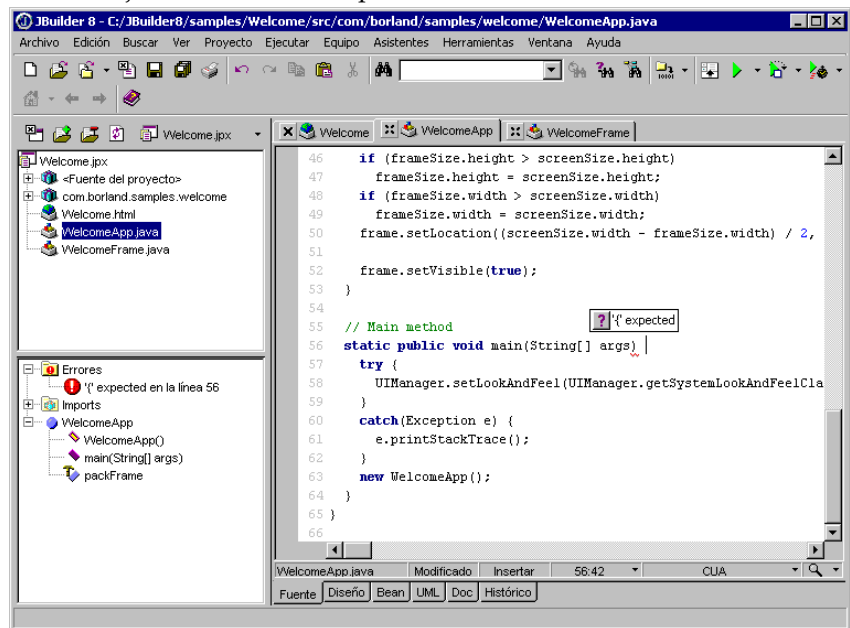
También se puede utilizar el panel de estructura para acceder a otras clases antecesoras e interfaces. Para ver el archivo `.java` de una clase antecesora o una interfaz, o el tipo de una variable que aparece en el panel de estructura, basta con hacer doble clic en el elemento o seleccionarlo y pulsar **Intro**. JBuilder muestra el archivo en los paneles de estructura y contenido. Si desea seleccionar un paquete, elija **Buscar | Buscar clases**, o bien haga doble clic en el paquete en el panel de estructura y seleccione

una clase en el cuadro de diálogo para abrirla en el panel de contenido. Para volver al archivo que se estaba visualizando, pulse el botón Inicio de la barra de herramientas principal.

Presentación de mensajes de error

Los errores de sintaxis aparecen subrayados en el editor. Coloque el cursor sobre el error para ver la ayuda inmediata y la naturaleza del error. La ayuda inmediata cuenta con uno de estos dos botones:

- Botón de lupa: púlselo para abrir ClassInsight.
- Botón de signo de interrogación: púlselo para abrir el archivo de ayuda de mensajes de error del compilador.



Los mensajes de errores de sintaxis se muestran en la carpeta Errores del panel de estructura. Amplíe la carpeta y seleccione un mensaje de error. La línea de código correspondiente aparece resaltada en el editor.

Consulte

- “Acerca de los mensajes de advertencia y error” en *Creación de aplicaciones con JBuilder*.

Panel de contenido

El panel de contenido muestra todos los archivos abiertos de un proyecto como un conjunto de pestañas. Para abrir un archivo en el panel de contenido, haga doble clic en él, en el panel del proyecto, o selecciónelo y pulse *Intro*. Los nombres de todos los archivos abiertos aparecen en las pestañas del panel de contenido. Cuando se pulsa la pestaña de un archivo, éste se convierte en el actual.

El panel de contenido permite también acceder a varias vistas y operaciones de archivo por medio de las pestañas de vistas de archivo, que se muestran en la parte inferior de las ventanas de archivo.



Pestañas de archivo

Las pestañas de archivo contienen los nombres de los archivos abiertos en un proyecto seleccionado en el panel de contenido. Sólo son visibles las pestañas correspondientes a los archivos abiertos en el proyecto activo. Sólo hay un archivo activo en cada editor abierto.



Las pestañas de archivo cuentan con un botón que tiene dos funciones:

- Si se pulsa, el archivo se cierra.
- Indica si el archivo ha cambiado desde que se guardó por última vez. Si es así, tiene el siguiente aspecto: , y se pregunta si se desea guardar el archivo antes de cerrarlo.

Las etiquetas de las pestañas se pueden personalizar de varias formas: orientación, tipo de etiqueta e inserciones. Para definir estas opciones, elija Herramientas | Opciones del IDE | Visualizador. Pulse el botón Ayuda del cuadro de diálogo si desea información sobre estas opciones.

Pestañas de vistas de archivo

Las pestañas que aparecen en la parte inferior del panel de contenido dependen del tipo de archivo abierto. Sólo se muestran las pestañas correspondientes al archivo abierto debajo de su ventana. Cada una de estas pestañas ofrece una vista diferente del archivo abierto. Por ejemplo, un archivo `.java` que se puede diseñar visualmente puede tener varias pestañas: Fuente, Diseño, Bean, UML, Doc e Histórico. Las pestañas disponibles varían según la versión de JBuilder.

Redimensionamiento del panel de contenido

El tamaño del panel de contenido se puede cambiar de las siguientes formas:

- Arrastre la barra de división que separa los paneles de proyecto y contenido para darle el tamaño deseado.
- Elija Ver | Ocultar todo en el menú principal o pulse *Ctrl+Alt+Z* si utiliza la asignación de teclas CUA, para ampliar al máximo el panel de contenido y ocultar los otros paneles.

Consulte

- “Menú Ver” si desea ver otras configuraciones de los paneles.

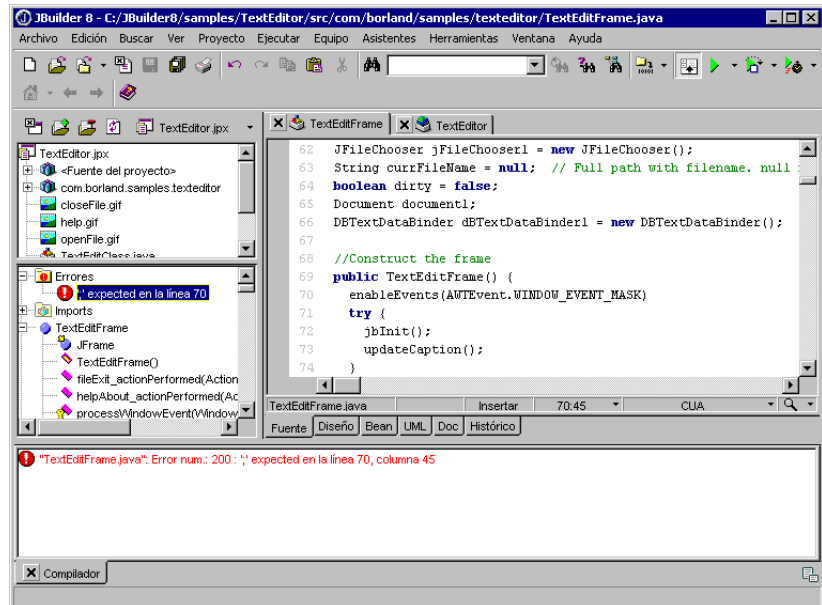
El panel de mensajes

En ocasiones, en la parte inferior del Visualizador de aplicaciones aparece un panel de mensajes con pestañas, que presenta jerárquicamente los mensajes de usuario de distintas operaciones, como:

- Generándose y compilándose
- Ejecutándose
- Depurando
- Buscando
- Perfeccionando código
- Comprobando unidad
- Control de versiones

JBuilder genera en la parte inferior del panel de mensajes una pestaña para cada proceso. Estas pestañas permiten interactuar con el proceso de ejecución. Un icono en la pestaña indica que hay un proceso activo y es posible la extracción de datos en modo de consola, por medio del área de texto del panel de mensajes.

A continuación se muestra un Visualizador de aplicaciones, con un panel de mensajes que informa de un error de sintaxis en la parte inferior:



Durante la compilación aparece la pestaña Generar si hay errores o advertencias.

Algunas de estas operaciones ofrecen en la pestaña botones de barra de herramientas que permiten detener o reanudar el proceso actual. El botón de reanudación asociado a un proceso utiliza los valores vigentes en el momento de la creación de la pestaña, y pasa por alto los cambios efectuados posteriormente. Esta función permite establecer distintas configuraciones para comprobarlas repetidamente.

Se pueden ejecutar varias operaciones de forma simultánea en el mismo proyecto o en proyectos distintos. Para cada operación que se inicia se crea una pestaña de mensaje, con los botones Depurar y Ejecutar de la barra de herramientas.

Para iniciar una operación de ejecución o depuración, asegúrese de que el proyecto que desea ejecutar o depurar es el activo en el panel del proyecto, o el archivo que desea depurar es el actual en el panel de contenido. A continuación, dé comienzo a la operación por medio de los botones de la barra de herramientas principal.

En algunas operaciones, como la ejecución, la depuración y el control de versiones, el panel de mensajes contiene una barra de estado que informa del progreso o el resultado de la operación actual.

Copia de texto del panel de mensajes

También es posible copiar el contenido del panel de mensajes. Haga clic con el botón derecho del ratón en el panel de mensajes y elija Copiar todo, o seleccione el texto que desea copiar y elija Copiar el contenido seleccionado.

Ocultación y presentación del panel de mensajes



Para mostrar u ocultar el panel de mensajes, elija Ver | Mensajes o pulse el botón Mensajes de la barra de herramientas para cambiar el estado. Si lo prefiere, haga clic en las pestañas de mensaje con el botón derecho del ratón y elija Ocultar vista de mensajes, o pulse *Ctrl+Alt+M* con la configuración de teclado CUA. Pulse el botón Cerrar pestaña para cerrar una pestaña en el panel de mensajes. Para eliminar todas las fichas del panel o algunas de ellas, haga clic con el botón derecho del ratón en las pestañas de mensaje y elija Eliminar pestaña <nombre> o Eliminar todas las pestañas.

Segregación del panel de mensajes

Si lo prefiere, puede segregar el panel de mensajes de modo que se convierta en una ventana flotante que se puede colocar en cualquier posición de la pantalla.



Para segregar el panel de mensajes, pulse el icono situado en la esquina inferior derecha del panel de mensajes.



Para volver a integrar el panel de mensajes en la parte inferior del Visualizador de aplicaciones, pulse el icono de la esquina inferior derecha del panel de mensajes flotante.

Depurador

La depuración es el proceso de búsqueda y solución de errores en los programas. La interfaz del depurador se encuentra en el panel de mensajes.

El depurador incluye estas funciones:

- Pestañas verticales que representan las vistas. Las vistas corresponden a la sesión de depuración seleccionada actualmente.
- Pestañas horizontales que representan las sesiones de depuración. Cada pestaña representa una sesión.
- La barra de herramientas del depurador corresponde a la sesión de depuración seleccionada actualmente.
- Una barra de estado, contigua a la barra de herramientas, que indica el estado del proceso de depuración.

Si el depurador no aparece después de seleccionar Ejecutar | Depurar proyecto, defina un punto de interrupción en una sentencia ejecutable haciendo clic en el lateral gris de la izquierda del editor, y elija de nuevo Ejecutar | Depurar proyecto.

La barra de herramientas de la parte inferior del depurador permite acceder rápidamente a los botones Terminar, Continuar, Pausa, Paso inteligente, Omitir inspección, Inspección, Abandonar inspección, Añadir punto de interrupción, Añadir punto de observación y Mostrar marco actual.

Consulte


- “Depuración de programas en Java” en *Creación de aplicaciones con JBuilder*.

Barras de estado

En el Visualizador de aplicaciones hay tres barras de estado.

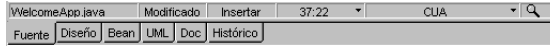
- Barra de estado principal
- Barra de estado de archivo
- Barra de estado de mensaje

La barra de estado principal se muestra en la parte superior de la ventana del Visualizador de aplicaciones, y le mantiene al corriente sobre las operaciones y su resultado.

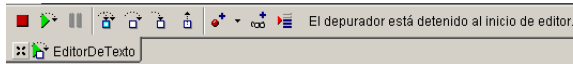


Activado el proyecto: TextEditor.jpr

La barra de estado de archivo aparece en la parte inferior de las ventanas de archivo abiertas del panel de contenido. Muestra información relacionada con el archivo actual, como el nombre, la posición del cursor (números de línea y columna), el modo de inserción si el archivo es de texto y el tamaño si el archivo es de imagen. Los archivos de texto también contienen una herramienta de lupa que se puede utilizar para cambiar el tamaño de las fuentes utilizadas en el editor.



Esta barra de estado aparece en la parte inferior del panel de mensajes durante procesos como la ejecución, la depuración y el control de versiones.



Desplazamiento y búsqueda con el Visualizador de aplicaciones

Utilice los siguientes métodos abreviados de teclado para desplazarse por los paneles del Visualizador de aplicaciones.

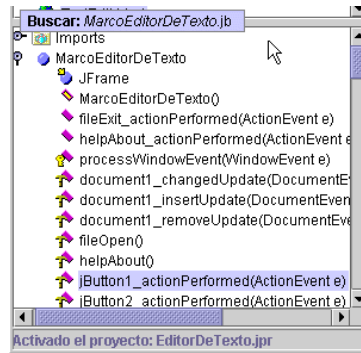
Tabla 4.2 Atajos de teclado para desplazamientos

Atajo de teclado	Acción
<i>Pestaña</i>	Permite desplazarse hacia adelante, en el orden de rotación, hasta el siguiente panel del Visualizador de aplicaciones.
<i>Mayús+Ctrl+Tab</i>	Permite retroceder, en el orden de rotación, hasta el panel anterior del Visualizador de aplicaciones.
Teclas <i>Flechas arriba/abajo</i>	Desplazan el cursor de selección hacia arriba o hacia abajo en un árbol.
<i>Flechas izquierda y derecha</i>	En los paneles de proyectos y estructura, amplían y contraen ramas de nodos de nivel superior de un árbol.
<i>Escriba</i>	En el panel del proyecto, abre un archivo de código fuente seleccionado. Es equivalente a hacer doble clic. En el panel de estructura, encuentra la superclase o interfaz de la clase seleccionada. Es equivalente a hacer doble clic.

Búsqueda en árboles

Para efectuar búsquedas en los paneles de proyecto, estructura y mensajes, active el panel y empiece a escribir. Se muestra un cuadro de ayuda inmediata en el que se puede definir un filtro de búsqueda. Por defecto, se selecciona la primera entrada que cumpla el criterio. Las entradas siguiente y anterior se seleccionan por medio de las flechas *Arriba* y *Abajo*. Pulse *Esc* para cancelar el modo de búsqueda. En la búsqueda no se tienen en cuenta los nodos contraídos del árbol.

Para buscar nodos contraídos, utilice la tecla de punto (.) o Control + punto (*Ctrl+.*) precedidas por el nombre del nodo. El nodo se amplía y la búsqueda se limita a sus descendientes. Por ejemplo, como se muestra a continuación, al insertar `EditorDeTexto.jb` en el panel de estructura selecciona el primer componente `JButton` en la arborescencia de `TextEditorFrame`.



Normalmente se seleccionan las entradas que comienzan por el texto especificado. Utilice caracteres comodín para ampliar la búsqueda:

- Un asterisco representa cualquier cadena de caracteres.
- Un signo de interrogación representa cualquier carácter.

Búsqueda de clases

Pulse la combinación de teclas *Ctrl+-* (signo menos) para abrir el cuadro de diálogo Buscar clases. En JBuilder Personal, con esta acción se abre el visualizador de clases utilizado normalmente. En JBuilder SE y Enterprise, cuenta con dos pestañas: *Buscar* y *Examinar*.

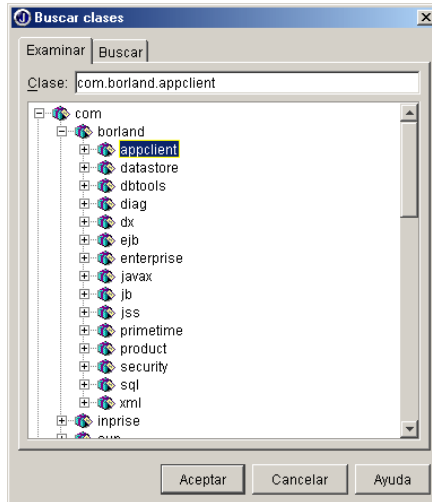
En la ficha *Buscar*:

- 1 Escriba en el campo *Buscar* para que JBuilder rellene el campo *Lista de correspondencias*.
- 2 Seleccione la clase que desea abrir.

3 Pulse Aceptar o *Intro*.

Se cierra el cuadro de diálogo, se abre la clase y se convierte en el archivo activo del editor.

La ficha Examinar presenta los paquetes y archivos fuente utilizados en el proyecto activo. Para ampliar un paquete, pulse el icono de ampliación contiguo o escriba el nombre completo de la clase que desea abrir. La rama relevante del árbol se amplía mientras se escribe:



1 Seleccione la clase que desea abrir.

2 Pulse Aceptar o *Intro*.

Se cierra el cuadro de diálogo, se abre la clase y se convierte en el archivo activo del editor.

Las dos fichas admiten los comodines * y ?.

Utilización de Favoritos

Los cuadros de diálogo de selección de archivos, como Añadir archivos o paquetes al proyecto, cuentan con enlaces rápidos a los directorios más utilizados. La ubicación de la lista Favoritos depende del sistema operativo. Por ejemplo, en Windows se encuentran en un panel a la izquierda del cuadro de diálogo, y en Mac, en una barra de navegación. Seleccione un favorito para colocar la selección del cuadro de diálogo en ese directorio.

JBuilder establece enlaces por defecto, pero, si así lo desea, puede añadir y personalizar sus propios enlaces.

Si desea añadir un enlace favorito desde un cuadro de diálogo de selección de archivos que esté abierto:

- 1 Desplácese hasta el archivo que desea añadir.



- 2 Pulse el icono Favoritos de la esquina superior derecha del cuadro de diálogo.

- 3 Seleccione Añadir a Favoritos

Se abre el cuadro de diálogo Añadir a Favoritos.

- 4 Escriba un nombre para la nueva vista.

- 5 Pulse Aceptar o *Intro*.

El nuevo favorito se añade a la lista. Los favoritos que añada se distinguen de los favoritos por defecto por el icono en forma de corazón. Si es necesario, se puede desplazar por la lista Favoritos.

Para organizar los favoritos añadidos mediante un cuadro de diálogo abierto de selección de archivos:

- 1 Pulse el icono Favoritos y elija Organizar Favoritos.

Aparece el cuadro de diálogo Organizar Favoritos.

- 2 Seleccione un favorito de la lista.

- 3 Cambie el nombre del favorito, bórrelo o desplácelo hacia arriba o abajo en la lista.

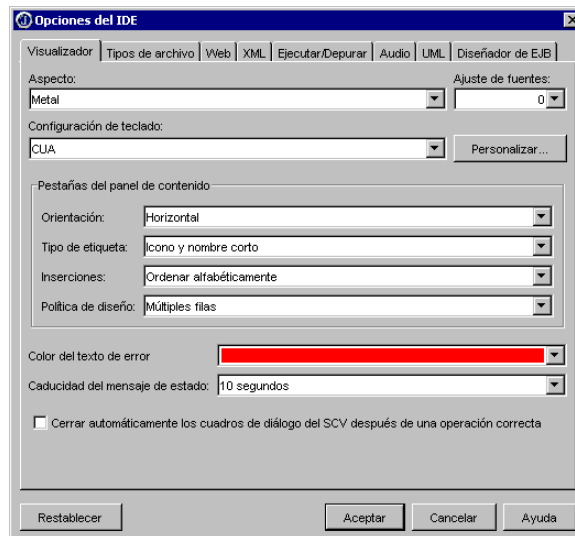
El orden de este cuadro de diálogo determina el de la lista de favoritos.

- 4 Pulse Aceptar o *Intro* cuando termine.

Personalización del IDE

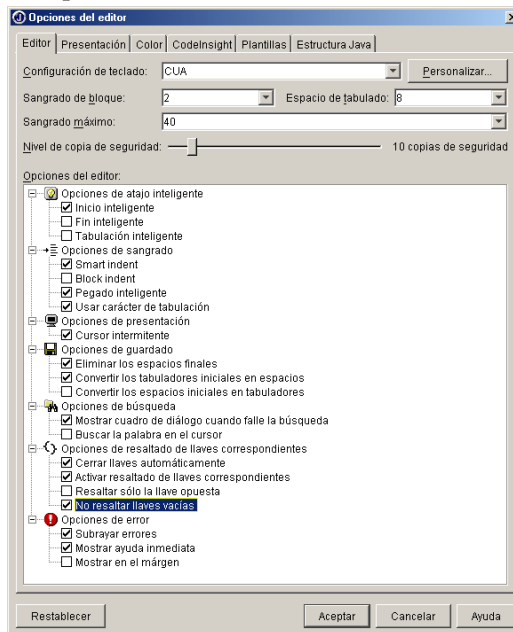
Existen diversas opciones para personalizar el aspecto y el comportamiento del IDE. Por ejemplo, es posible elegir el aspecto del Visualizador de aplicaciones, seleccionar o personalizar la configuración de teclado que determina las teclas que se utilizan para acceder al IDE y especificar cómo aparecen las pestañas de los archivos del panel de contenido. Existen opciones adicionales para determinadas partes del IDE, tales como Ejecutar/Depurar y UML. Todas estas opciones están disponibles en el cuadro de diálogo Opciones del IDE.

Para acceder al cuadro de diálogo Opciones del IDE, seleccione Herramientas | Opciones del IDE.



Si desea más información sobre las diversas opciones disponibles en el cuadro de diálogo Opciones del IDE, pulse el botón Ayuda.

Para personalizar el editor, seleccione Herramientas | Opciones del editor.



Pulse el botón Ayuda si desea información sobre las opciones del editor.

El editor

JBUILDER posee un magnífico editor de código con numerosas funciones que facilitan y aceleran la tarea de escribir código.

Para abrir un archivo en el editor, haga doble clic en un archivo de texto, en el panel del proyecto (arriba a la izquierda), o selecciónelo y pulse *Intro*. Observe la barra de estado de archivo de la parte inferior del editor, que indica el nombre de archivo, la posición del cursor (número de línea y columna) y el modo de inserción del archivo de texto.

El editor ofrece una amplia gama de funciones de productividad, como la correspondencia de llaves de apertura y cierre, el resaltado de sintaxis, los atajos de teclado, la personalización de la configuración de teclado del editor, CodeInsight, las plantillas de código, la búsqueda y la impresión, así como un editor completamente personalizable. Muchas de estas funciones se pueden definir en el cuadro de diálogo Opciones del editor (Herramientas | Opciones del editor).

Existen muchas más funciones en el menú contextual del editor. Éstas varían según la edición de JBUILDER que se utiliza, la configuración del proyecto y la posición del cursor en el editor cuando se abre el menú. Haga clic con el botón derecho del ratón con el cursor en el editor o haga clic sobre el editor y pulse *Mayús+F10* para desplegar el menú contextual.

Selección de una configuración de teclado para el editor

Los atajos o configuraciones de teclado realizan determinadas tareas de forma mucho más rápida y sencilla. Los diferentes editores asignan diferentes combinaciones de teclas a la misma acción. Por ello, los cambios de editor pueden resultar incómodos y arriesgados.

Dado que cada programador tiene sus hábitos y preferencias, JBuilder ofrece varias emulaciones de editor. Elija la que le resulte más cómoda:

- BRIEF
- CUA
- Emacs
- Macintosh
- Macintosh CodeWarrior
- Visual Studio

Cuando se instala JBuilder para Windows, Linux o Solaris, la configuración de teclado activa es CUA. En el caso de JBuilder para Macintosh, la configuración activa es Macintosh.

Para seleccionar otro esquema de configuración de teclado para el editor:

- 1 Pulse la flecha abajo, situada a la izquierda de la lupa, en la parte derecha de la barra de estado.
- 2 Seleccione el esquema de configuración de teclado que desee.

Si desea cambiar el esquema de configuración de teclado, también puede utilizar los cuadros de diálogo Opciones del IDE y Opciones del editor.

- 1 Seleccione Herramientas | Opciones del IDE y abra la ficha Visualizador; o bien, seleccione Herramientas | Opciones del editor y abra la ficha Editor.
- 2 Haga clic en el campo Configuración de teclado y, en el menú desplegable, seleccione la emulación de editor que desee.
- 3 Pulse Aceptar.

La nueva emulación de teclado se activa inmediatamente.

Personalización de las definiciones de teclado

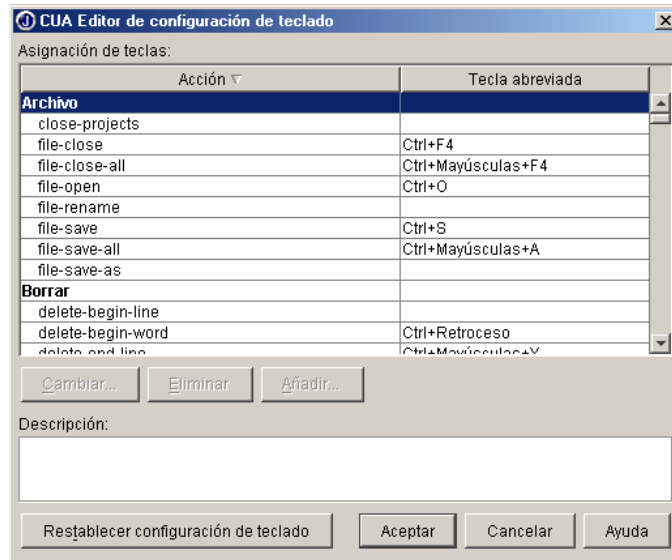
Si posee JBuilder SE o Enterprise, puede personalizar la configuración de teclado seleccionada mediante el Editor de configuración de teclado. En JBuilder Personal, el Editor de configuración de teclado es de sólo lectura; las configuraciones de teclado no se pueden modificar.

Para tener acceso al Editor de configuración de teclado:

- 1 Seleccione Herramientas | Opciones del IDE y abra la ficha Visualizador; o bien, seleccione Herramientas | Opciones del editor y abra la ficha Editor.

- 2 Pulse el botón Personalizar, contiguo al campo Configuración de teclado.
- 3 Seleccione el comando que desea modificar o eliminar, o al que desea añadir una configuración.
- 4 Pulse el botón correspondiente a la acción que desea realizar: Cambiar, Eliminar o Añadir.
- 5 Pulse sobre las cabeceras de la tabla para cambiar el orden de clasificación.

Figura 5.1 Editor de configuración de teclado CUA



Si desea más información sobre el uso del Editor de configuración de teclado, pulse su botón Ayuda.

División de la vista del código fuente

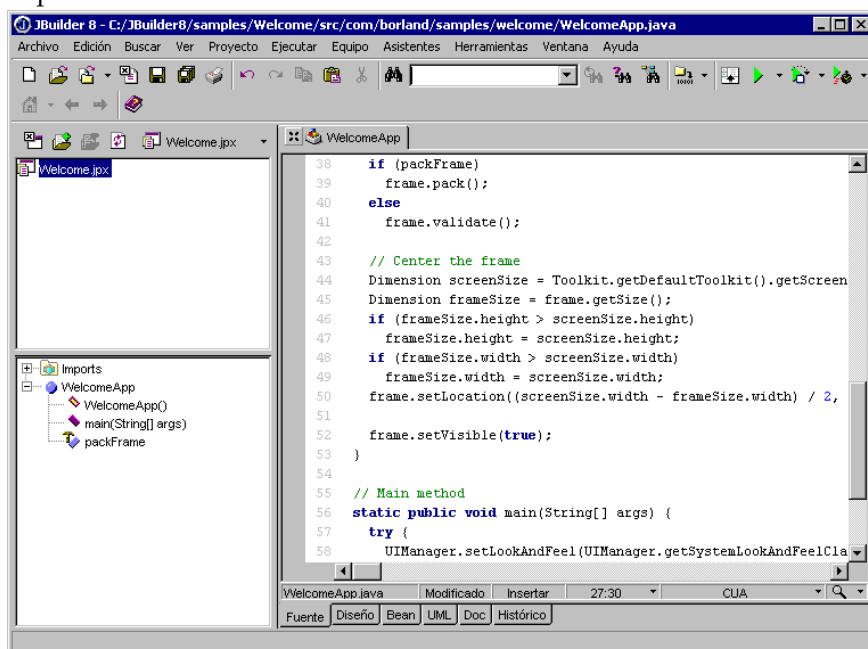
Es una función de
JBuilder SE y Enterprise.

El editor permite dividir la presentación del código fuente de un archivo en dos o más paneles verticales u horizontales. Esta configuración se aplica a un solo archivo, y no a todos los archivos abiertos del panel de contenido. Es posible asignar una configuración distinta a cada archivo abierto.

Para dividir la vista, haga clic con el botón derecho del ratón en el panel Fuente y elija Dividir en vertical o Dividir en horizontal. Para volver a la presentación en un panel, haga clic con el botón derecho del ratón en los paneles y elija Cerrar vista.

Números de línea

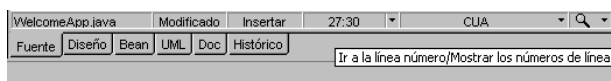
El editor presenta la opción de mostrar los números de línea en el margen izquierdo.



Esta función se puede activar o desactivar mediante la opción Numerar líneas de la ficha Editor del cuadro de diálogo Opciones del editor (Herramientas | Opciones del editor). También es posible desplazar instantáneamente el cursor hasta un determinado número de línea. Para ello, seleccione la opción de menú Buscar | Ir a la línea e indique el número de línea. También es posible acceder a estas dos funciones mediante los atajos disponibles en la barra de estado.

Presentación de los números de línea

La numeración de las líneas se puede controlar rápidamente desde el área de estado, en la parte inferior del editor. Los números de línea y columna de la posición del cursor se muestran siempre en la barra de estado. A la derecha de los números existe una flecha abajo que abre un menú desplegable.

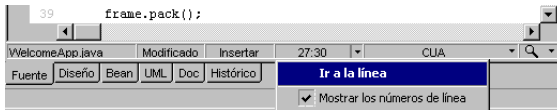


Para mostrar los números de línea, pulse la flecha abajo y active la opción **Mostrar los números de línea** en el menú desplegable. Para ocultarlos, pulse la flecha abajo y desactive la opción **Mostrar los números de línea** en el menú desplegable.



Desplazamiento a una línea determinada

Para ir a un número de línea determinado, haga clic en el número de línea o columna, o pulse la flecha abajo situada al lado y seleccione **Ir a la línea**; a continuación, en el cuadro de diálogo **Ir a la línea**, especifique el número de línea deseado.



También es posible abrir el cuadro de diálogo **Ir a la línea** mediante la combinación de teclas **Ctrl+G**.

Selección de texto

Existe una gran variedad de maneras de seleccionar texto en el editor. Para seleccionar una sección de texto, arrastre con el ratón; o bien, coloque el cursor en la posición inicial y mantenga pulsada la tecla **Mayús** mientras pulsa el botón del ratón al final del área de selección. También puede utilizar **Ctrl+Mayús** junto con las teclas de flechas para resaltar palabras y líneas con rapidez.

Además, existen atajos de teclado en el editor que le permiten seleccionar líneas completas de texto mediante el número de línea que aparece en el margen lateral.

Nota Para utilizar estos métodos abreviados, asegúrese de que los números de línea aparecen en el margen lateral izquierdo (consulte el apartado [“Presentación de los números de línea” en la página 5-4](#)).

Selección de una línea entera

Para seleccionar toda una línea, haga clic en su número.

Selección de un bloque de números de línea

Es posible seleccionar un bloque de líneas contiguas sin necesidad de desplazarse por la pantalla. Haga clic en el número de la primera línea, en el margen lateral, y mantenga pulsada la tecla *Mayús* mientras hace clic en el número de la última línea.

Selección de todos los números de línea

Para seleccionar todas las líneas de un archivo, coloque el cursor en el margen superior de un número de línea y pulse *Ctrl* mientras hace clic con el ratón sobre el número de línea.

Desplazamiento de texto mediante la operación de arrastrar y soltar

El editor permite arrastrar el texto seleccionado a otra ubicación. Resalte el texto deseado y arrástrelo con el ratón hasta su nueva posición. Si mueve una línea completa de texto, y la opción Pegado inteligente se encuentra seleccionada en Opciones del editor, la línea recibirá automáticamente el sangrado apropiado cuando la coloque en su nueva posición.

Cambio de tamaño de la fuente utilizada en el editor

El editor permite cambiar rápidamente el tamaño de la fuente del código.

Para aumentar el tamaño de las letras del código, haga clic en el icono de lupa situado en la parte derecha de la barra de estado del editor. Cada vez que pulse el icono, el texto aumentará de tamaño. También puede incrementar el tamaño de la fuente si pulsa sobre la flecha del menú desplegable situado junto al icono de lupa y selecciona Ampliar.

Para reducir la letra del código, pulse la flecha del menú desplegable situado junto al icono de lupa y seleccione Reducir.

Para restablecer el tamaño del texto del editor a su valor por defecto, pulse la flecha del menú desplegable situado junto al icono de lupa y seleccione Normal.

También puede cambiar tanto la fuente como su tamaño mediante el cuadro de diálogo Opciones del editor (Herramientas | Opciones del editor). Pulse la pestaña Presentación, realice los cambios deseados y utilice las opciones de Fuente del editor de esa ficha. Si desea más información, pulse el botón Ayuda del cuadro de diálogo.

Si desea información sobre la forma de cambiar el tamaño de la fuente utilizada en todo el IDE, incluido el editor, seleccione Herramientas | Opciones del IDE y pulse el botón Ayuda; de este modo, podrá leer información sobre la opción Aumento del tamaño de fuente, que aparece en la ficha Visualizador de este cuadro de diálogo.

Búsqueda de texto en el editor

El editor ofrece numerosos métodos de búsqueda y sustitución de texto concreto en archivos. Los comandos de búsqueda se encuentran en el menú Buscar y en los iconos de la barra de herramientas principal. Las opciones de búsqueda se pueden modificar en la ficha Editor del cuadro de diálogo Opciones del editor (Herramientas | Opciones del editor | Editor).

Tabla 5.1 Comandos de búsqueda de texto

Tarea	Comando
Busca texto	Buscar Buscar
Busca texto en todos los archivos de las vías de acceso especificadas	Buscar Buscar en vía de acceso
Busca una cadena de texto y la sustituye por otra	Buscar Reemplazar
Vuelve a buscar el mismo texto	Buscar Buscar de nuevo
Busca texto de forma incremental, a medida que se escribe la cadena de búsqueda.	Buscar Búsqueda incremental
Va a una línea con un número concreto	Buscar Ir a la línea
Busca en una clase, una interfaz o un paquete	Buscar Buscar clases
Busca la declaración de una variable, un método, una clase o una interfaz	Buscar Buscar definición
Busca usos o instancias de una variable, un método, una clase o una interfaz	Buscar Buscar referencias
Se desplaza a un elemento concreto de la lista	Ver Histórico
Vuelve al elemento anterior de la lista	Ver Atrás
Va al elemento siguiente de la lista	Ver Adelante
Define marcadores y regresa a ellos	Combinaciones de teclas del archivo de configuración del teclado (Ayuda Configuraciones de teclado)

Los cuadros de diálogo Buscar y Reemplazar ofrecen la posibilidad de reemplazar texto. Cuentan con otras opciones, como la búsqueda con distinción entre mayúsculas y minúsculas, el uso de comodines y expresiones regulares, y la definición del alcance de la tarea de búsqueda y sustitución. Pulse Ayuda en esos cuadros de diálogo si desea más información sobre el uso estas opciones.

Búsqueda de definiciones de símbolos

Es una función de JBuilder SE y Enterprise.

El comando Buscar definición, disponible en el menú contextual del editor y en el menú Buscar, permite desplazarse del uso de un símbolo a su definición. Para que sea posible buscar la definición de un símbolo es necesario que el proyecto se encuentre compilado y que la clase que contiene la definición se encuentre en la vía de acceso `import`

Para ver la definición de un símbolo, haga clic en él con el botón derecho del ratón y seleccione Buscar definición.

- 1 Coloque el cursor en el editor, sobre el símbolo cuya definición desea ver.

- 2 Haga clic con el botón derecho en el símbolo y elija Buscar definición.

En el editor se abre el archivo fuente donde se define el símbolo, con el cursor sobre la definición.

Búsqueda de referencias a símbolos

Es una función de JBuilder SE y Enterprise.

El comando Buscar referencias, disponible en el menú contextual del editor y en el menú Buscar, permite localizar todos los archivos fuente que utiliza un símbolo determinado.

Si desea buscar todas las referencias a un símbolo, haga clic en él con el botón derecho del ratón y seleccione Buscar referencias.

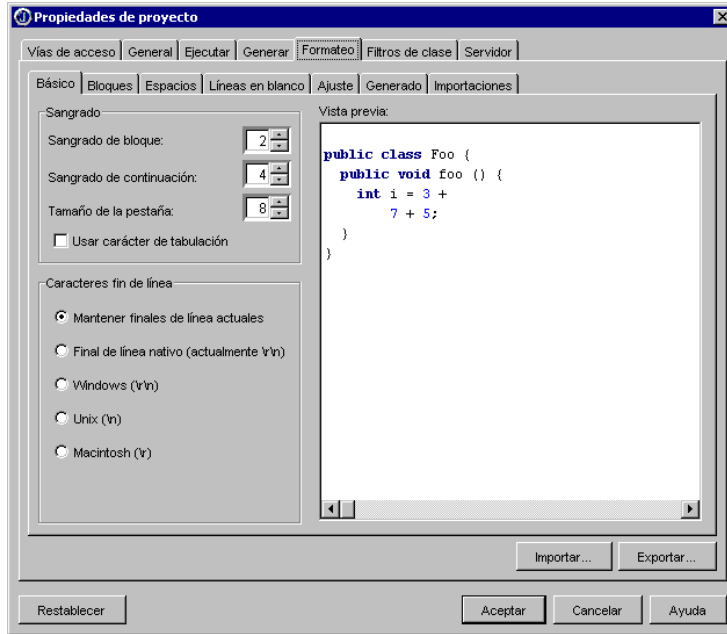
Si desea buscar los archivos fuente donde se encuentran las referencias, amplíe un nodo de categoría y desplácese por el árbol de jerarquía, en el panel de mensajes. Haga doble clic en una referencia para abrir el archivo fuente y colocar el cursor directamente sobre ella.

Importante

Para que sea posible buscar referencias, el proyecto debe estar compilado. Si desea más información, consulte “Configuración para la detección y el perfeccionamiento de referencias” en *Creación de aplicaciones con JBuilder*.

Cómo aplicar formato al código

Ahora, JBuilder le ofrece la posibilidad de definir sus preferencias de formato y aplicar formato automáticamente al código fuente. Para acceder a esas opciones de configuración, seleccione Proyecto | Propiedades de proyecto y pulse la pestaña Formateo.



Las pestañas de esta ficha permiten personalizar el formato del código mediante los siguientes tipos de preferencias:

- Sangrado
- Espacio de tabulado
- Caracteres fin de línea
- Parámetros de multilínea
- Sangrado de continuación
- Llaves
- Espacios
- Líneas en blanco
- Ajuste de texto
- Tratamiento de sucesos
- Sentencias de importación y su orden

Todas las fichas cuentan con una ventana Vista previa que muestra el aspecto de la preferencia seleccionada.

Aplicación de formato al código fuente

Para aplicar formato al código, abra un archivo en el editor y seleccione Edición | Formato o pulse la tecla Tab.

Importante Las propiedades de formato se aplican a todos los archivos del proyecto.

También puede exportar las preferencias de formato, o importar unas preferencias guardadas previamente, mediante los botones Exportar e Importar, respectivamente, situados en la parte inferior de la pestaña Formato del cuadro de diálogo Propiedades de proyecto.

Si desea más detalles sobre la configuración de opciones de formato de código, pulse el botón Ayuda, en la parte inferior de la ficha Formato del cuadro de diálogo Propiedades de proyecto.

Colocación de llaves en los bloques de código

Para colocar un bloque de código entre llaves, sitúe una llave de apertura al principio del bloque, desplácese hasta el final del bloque y pulse *Intro*.

Métodos abreviados Javadoc del editor

Javadoc es una utilidad de Sun Microsystems que genera archivos de documentación HTML a partir de los comentarios introducidos en los archivos fuente de la API. Los comentarios deben tener un formato acorde con la norma Javadoc. El Asistente para Javadoc de JBuilder (Asistentes | Asistente para Javadoc) permite configurar las propiedades de creación del Javadoc (JBuilder SE y Enterprise). Cuando se crea el proyecto, el Javadoc se genera automáticamente.

Con el fin de facilitar la codificación de comentarios Javadoc, el editor de JBuilder cuenta con una plantilla de comentarios Javadoc, que se activa cuando se escribe `/**` y se pulsa *Intro*. La plantilla añade automáticamente el símbolo `*/`, que indica el final del comentario Javadoc. Si el cursor está colocado inmediatamente antes de una clase, una interfaz o una firma de método, la plantilla se amplía para incluir las etiquetas Javadoc correspondientes (JBuilder SE y Enterprise).

Para añadir rápidamente un bloque de comentarios Javadoc:

- Coloque el cursor en el nivel adecuado de sangrado antes de una clase, una interfaz o una firma de método.
- Escriba `/**`.
- Pulse *Intro*.

En JBuilder SE y Enterprise, el editor añade automáticamente el símbolo de final de comentario Javadoc y coloca el cursor en la segunda línea del comentario. Si es necesario, también añade las etiquetas adecuadas.

Por ejemplo, si se escribe `/**` antes de las sentencias de importación en un archivo fuente de clase, se crea el siguiente bloque de comentarios:

```
/**
 * <p>Título: </p>
 * <p>Descripción: </p>
 * <p>Copyright: Copyright (c) 2001</p>
 * <p>Empresa: </p>
 * @author
 * @versión 1.0
 */
```

En JBuilder Personal, el editor añade automáticamente el símbolo de final de comentario Javadoc y coloca el cursor en la segunda línea del comentario.

En JBuilder SE y Enterprise, cuando se escribe `/**` para la siguiente firma de método:

```
public void addValues(Double valueOneDouble, Double valueTwoDouble)
```

se crea el siguiente comentario Javadoc:

```
/**
 *
 * @param valueOneDouble
 * @param valueTwoDouble
 */
```

Si desea más información acerca de JBuilder y Javadoc, consulte “Creación de Javadoc a partir de archivos fuente del API” en *Creación de aplicaciones con JBuilder*.

Utilización de etiquetas @todo en el editor

Las etiquetas `@todo` de Javadoc resultan útiles para recordar que es necesario hacer algo en una zona del código. Estas etiquetas se colocan dentro de los comentarios Javadoc.

Es una función de JBuilder SE y Enterprise: estas etiquetas `@todo` aparecen en el panel de estructura de JBuilder en la carpeta *Por Hacer*.

Las plantillas de código de JBuilder facilitan enormemente la adición de etiquetas `@todo` al código.

- 1 Escriba `todo` en el nivel adecuado de sangrado del editor.
- 2 A continuación pulse `Ctrl+J` para ampliar la plantilla en el código.

```
/** @todo <cursor situado aquí> */
```

Algunos asistentes de JBuilder generan etiquetas `@todo` para recordar que es necesario añadir código al stub generado

Compatibilidad de impresión en el editor

El comando Archivo | Imprimir permite imprimir el código fuente directamente desde el editor.

El comando Archivo | Configuración de página muestra el cuadro de diálogo del mismo nombre, donde se pueden definir las opciones de diseño:

- Configuración de página: Las opciones de la ficha Configuración de página permiten definir la orientación, el diseño y la fuente de la página, además de configurar la numeración y el ajuste de líneas.
- Avanzada: Las opciones de la ficha Avanzada permiten establecer los márgenes y encabezados de impresión: Las siguientes variables permiten controlar lo que se imprime en el encabezado.

Tabla 5.2 Variables del encabezado

Variable	Descripción
%f	Nombre de archivo y vía de acceso
%g	Sólo nombre de archivo
%p	Número de página
%n	Número total de páginas
%t	Tiempo
%d	Fecha (formato largo)
%s	Fecha (formato corto)
%u	Nombre de usuario

Arrastrado de archivos al editor

Para abrir un archivo externo a JBuilder en el editor, puede utilizar la técnica de arrastrar y soltar con el ratón. El archivo debe ser de un tipo reconocido por JBuilder. Aunque el archivo se abre en el editor, no se añade al proyecto. Si desea añadir el archivo al proyecto, utilice el botón Añadir archivos/paquetes que se encuentra en la parte superior del panel del proyecto o haga clic con el botón derecho del ratón sobre el archivo del proyecto, en el panel del proyecto, y seleccione Añadir archivos/paquetes. La técnica de arrastrar y soltar es una opción que se puede desactivar en el cuadro de diálogo Opciones del editor.

Atajos de codificación

JBuilder incluye CodeInsight para ayudar en la elaboración del código, y plantillas que permiten insertar rápidamente elementos de código de uso frecuente.

CodeInsight

CodeInsight ofrece rellenado del código, listas de parámetros y ayuda inmediata para la evaluación de expresiones dentro de archivos Java y dentro de los segmentos de código de los archivos JSP. CodeInsight de JBuilder presenta en el editor ventanas emergentes sensibles al contexto que muestran lo siguiente:

- MemberInsight: Miembros de datos y métodos accesibles para el contexto actual.
- ClassInsight: Clases a las que se puede acceder por la vía de acceso a clases actual.
- ParameterInsight: Parámetros esperados por los métodos que se están codificando.
- Buscar definición: Profundiza en el código fuente en busca de la variable, el método o la clase actual en el editor.
- Evaluación inmediata de expresiones: Los valores de las variables aparecen en el depurador.
- ExpressionInsight: El contenido de la expresión seleccionada en el depurador.

Es una función de JBuilder SE y Enterprise.

Es una función de JBuilder SE y Enterprise.

Es una función de JBuilder SE y Enterprise.

Importante

Para que las clases importadas estén disponibles para CodeInsight es necesario que el proyecto esté compilado.

Figura 5.2 Ventana emergente de CodeInsight



Abra la herramienta CodeInsight, seleccione el símbolo que desea insertar y pulse *Intro*.

Para acceder a CodeInsight, utilice estos métodos abreviados de teclado:

Tabla 5.3 Teclas de método abreviado para CodeInsight

Ventana emergente	Teclas abreviadas	Acciones
MemberInsight	<i>Ctrl+H</i>	Proporciona métodos y miembros de ámbito activo.
ParameterInsight	<i>Ctrl+Mayús+H</i>	Ofrece parámetros de métodos.
Buscar símbolo	<i>Ctrl+Intro</i>	Profundiza en busca de la definición de un símbolo.
ClassInsight	<i>Ctrl+Alt+Espacio</i> <i>Ctrl+Alt+H</i>	Inserta un nombre de clase en un archivo modificable. JBuilder Enterprise ofrece distintos estilos de inserción.
Plantillas de código	<i>Ctrl+J</i>	Ofrece plantillas de código.

Nota Si utiliza Macintosh, pulse la tecla *Comando* en lugar de *Ctrl*.

Estas teclas de método abreviado son las opciones por defecto para JBuilder. Si desea ver la lista actual de pulsaciones abreviadas de alguna configuración de teclado, consulte el cuadro de diálogo Editor de configuración de teclado. Si desea abrir el Editor de configuración de teclado, seleccione Herramientas | Opciones del editor, elija la configuración que desee ver y pulse el botón Personalizar de la ficha Editor. En JBuilder SE y Enterprise se puede configurar la asignación de teclas con este cuadro de diálogo.

Sugerencia Si CodeInsight falla, busque errores como la falta de llaves y sentencias de importación.

Pulse las teclas de flecha para desplazar la selección hacia arriba y abajo en la lista. Para aceptar una selección de CodeInsight, pulse *Intro*, o cualquier carácter que no sea alfanumérico ([, =, \, etc.) o la barra espaciadora.

Consulte

- Ayuda | Configuraciones de teclado
- [“Personalización del editor” en la página 5-20.](#)
- “Búsqueda de definiciones de símbolos” en *Creación de aplicaciones con JBuilder*.

MemberInsight

MemberInsight presenta una lista emergente de todas las llamadas a métodos adecuadas para un prefijo de referencia determinado, según la configuración de CodeInsight. La lista emergente aparece automáticamente cuando se escribe un carácter de punto (.) en un contexto válido. Esta lista también se puede abrir de forma manual, mediante la pulsación de *Ctrl+Espacio* o *Ctrl+H*.

También se puede especificar el nombre de una clase para obtener los nombres de los métodos, propiedades y sucesos que se pueden emplear con ella. Esta función también se puede emplear para completar sentencias de asignación. CodeInsight también completa el código automáticamente. Si el resto de la sentencia de código no se repite, CodeInsight lo rellena todo automáticamente, y la lista de miembros no aparece.

La lista de miembros se basa en las importaciones del proyecto actual. Los métodos desaconsejados se señalan por medio del tachado. La lista también se filtra según las reglas de accesibilidad de Java.

ClassInsight

En JBuilder Personal, ClassInsight es el visualizador de clases más utilizado. Para abrir ClassInsight, pulse *Ctrl+Alt+Espacio*, o bien, pulse *Ctrl+Alt+H* en el editor y busque la clase que desea añadir. La sentencia de importación se añade automáticamente.

Es una función de
JBuilder SE y Enterprise.

En JBuilder SE y Enterprise, ClassInsight abre el cuadro de diálogo ClassInsight, que se puede utilizar para insertar clases en el código. Para abrir el cuadro de diálogo, pulse *Ctrl+Alt+Espacio* o *Ctrl+Alt+H* en el editor. Tiene dos fichas: Buscar y Examinar. La ficha Buscar ofrece un campo de búsqueda y una lista de coincidencias para lo que escriba. La ficha Examinar cuenta con un campo de búsqueda y un examinador de clases.

Comience a escribir en el campo Buscar el nombre de la clase que desea insertar en el código. Mientras se escribe, JBuilder busca de forma dinámica las clases en la vía de acceso a clases actual, y muestra una lista de posibles coincidencias, que varía a medida que se sigue escribiendo.

Pulse *Intro* o haga doble clic en la clase seleccionada para insertarla y, a continuación, pulse *Ctrl+Intro* para insertar el paquete. También puede utilizar las teclas de flecha para desplazarse por la lista. Para que sea posible encontrar clases en todas las bibliotecas, éstas deben encontrarse en el proyecto. Cuando se añade una clase por medio de ClassInsight, la última sentencia de importación se añade automáticamente al código. Para reordenar y optimizar las sentencias de importación, elija Optimizar sentencias de importación en el editor. Consulte “Optimización de importaciones” en *Creación de aplicaciones con JBuilder*.

Importante Si ha añadido un archivo fuente o una biblioteca al proyecto, para que ClassInsight reconozca las adiciones es necesario volver a compilar el proyecto y pulsar el botón Actualizar del panel del proyecto. Elija Proyecto | Ejecutar Make del proyecto o Proyecto | Generar el proyecto para repetir la compilación.

Consulte

- “Optimización de importaciones” en *Creación de aplicaciones con JBuilder*.
- “Adición y configuración de bibliotecas” en *Creación de aplicaciones con JBuilder*.
- “Vía de acceso a clases” en *Creación de aplicaciones con JBuilder*.

Opciones de inserción

Es una función de JBuilder SE y Enterprise.

ClassInsight cuenta con varias opciones de inserción que permiten controlar la forma en que el nombre de la clase y la sentencia de importación se insertan en el código. Los ejemplos siguientes utilizan la clase BorderLayout del paquete `java.awt`.

• Insertar nombre corto de clase con sentencia de importación

Inserta el nombre corto de la clase con una sentencia de importación que incluye el paquete completo y el nombre de la clase. Por ejemplo:

- Nombre de clase insertado: BorderLayout
- Sentencia de importación insertada: `import java.awt.BorderLayout;`

• Insertar nombre corto de clase con sentencia de importación de paquete

Inserta el nombre corto de la clase con la sentencia de importación del paquete. Por ejemplo:

- Nombre de clase insertado: BorderLayout
- Sentencia de importación insertada: `import java.awt.*;`

- **Insertar nombre completo de clase**

Inserta el nombre completo de la clase (paquete + nombre de la clase) sin sentencia de importación. Por ejemplo:

- Nombre de clase insertado: `java.awt.BorderLayout`
- Sentencia de importación insertada: ninguna

Consulte

- “Optimización de importaciones” en *Creación de aplicaciones con JBuilder*.

ParameterInsight

Cuando se crea el código para una llamada a un método se puede mostrar una lista de los parámetros esperados para el método. Para ello, escriba el nombre del método y pulse **Ctrl+Mayús+Espacio**, **Ctrl+Mayús+H** o el carácter de paréntesis de apertura (`()`). Se muestran todos los parámetros posibles, incluidos los sobrecargados.

Si está disponible el código fuente del método, se muestran los nombres de los parámetros. A medida que se introducen los parámetros de la llamada al método, el parámetro actual se muestra resaltado en la lista.

Buscar definición (Profundizar)

Es una función de JBuilder SE y Enterprise.

Buscar definición permite dirigirse rápidamente al código fuente de la variable, el método o la clase seleccionado actualmente en el editor, si existe. Si no está disponible conduce al código fuente del stub. Para utilizar esta función, coloque el cursor en el símbolo, en el código fuente, y pulse **Ctrl+Intro**. Si desea ver una lista completa de las combinaciones de teclas por configuraciones de teclado, consulte Ayuda | Configuración de teclado.

Evaluación inmediata de expresiones

Es una función de JBuilder SE y Enterprise.

Si está depurando un programa y coloca el cursor sobre una variable, en el editor, se muestra su valor. El valor aparece en una pequeña ventana emergente de aspecto similar a una ventana de ayuda inmediata.

ExpressionInsight

Es una función de JBuilder SE y Enterprise.

Interrumpe el depurador y coloca el cursor dentro de una expresión para acceder a ExpressionInsight. ExpressionInsight es una pequeña ventana emergente que muestra el contenido de la expresión seleccionada en una estructura de árbol.

Para mostrar la ventana ExpressionInsight:

- Mantenga pulsado el botón *Ctrl* (botón *Comando* en Macintosh) y lleve el ratón sobre el código en el editor. La ventana ExpressionInsight se muestra cuando el ratón pasa sobre una expresión con sentido.
- Mueva el ratón a la expresión que se desea analizar en más detalle y pulse *Ctrl* junto con el botón derecho del ratón.

Consulte

- “Depuración de programas en Java” en *Creación de aplicaciones con JBuilder*.

Configuración de CodeInsight

Para configurar CodeInsight, haga clic con el botón derecho del ratón en el panel Fuente, elija Opciones del editor y abra la ficha CodeInsight (o elija Herramientas | Opciones del editor | CodeInsight). Si desea más información, pulse el botón Ayuda de la ficha CodeInsight del cuadro de diálogo.

JBuilder proporciona varias plantillas de código predefinidas. Seleccione Herramientas | Opciones del editor | Plantillas para ver las disponibles.

Plantillas de código

La adición, la modificación y el borrado de plantillas son funciones de JBuilder SE y Enterprise. Las plantillas de código son de sólo lectura en JBuilder Personal.

Las plantillas de código son fragmentos de elementos de código de uso frecuente que se pueden insertar en el código para ahorrar tiempo. Puede utilizar las plantillas de código en el editor para agilizar el proceso de escritura.

Para utilizar una plantilla de código, seleccione uno de los siguientes métodos:

- Escriba el nombre de la plantilla de código, como `classp`, en el lugar en que desea insertarla y pulse *Ctrl+J*. Para ver los nombres de las plantillas, consulte Herramientas | Opciones del Editor | Plantillas.
- Coloque el cursor en el lugar en el que debe aparecer el código y pulse *Ctrl+J*. Se abrirá una lista de plantillas de código. Elija una de ellas, desplazándose con las teclas de flecha, y pulse Intro.

El editor amplía la plantilla automáticamente.

Sugerencia

Una vez que ha desplegado la plantilla, utilice CodeInsight para que le ayude a escribir el código. Consulte “[Atajos de codificación](#)” en la [página 5-13](#).

Las plantillas de código predefinidas de JBuilder tienen el formato elegido en las preferencias del proyecto. Estas opciones se configuran en los siguientes lugares:

- Opción Llaves: Proyecto | Propiedades de proyecto | Estilo de código (ficha)
- Opción Sangrado de bloque: Herramientas | Opciones del editor | Editor (ficha)

Modificar plantillas de código

Es una función de
JBuilder SE y Enterprise.

Es posible modificar estas plantillas y borrarlas si no son necesarias. También se pueden añadir plantillas propias. Para modificar esta lista, haga clic con el botón derecho del ratón en el editor, elija Opciones del editor y abra la pestaña Plantillas.

Para crear una plantilla de código:

- 1 Elija Herramientas | Opciones del editor y pulse la pestaña Plantillas.
- 2 Pulse el botón Añadir de la ficha Plantillas.
- 3 Escriba el nombre y la descripción de la plantilla de código y pulse Aceptar.
- 4 Escriba el código de la nueva plantilla en el área de edición Código.
- 5 Añada y modifique las plantillas deseadas, y cuando termine, pulse Aceptar para cerrar el cuadro de diálogo Opciones del editor.

Para modificar una plantilla de código:

- 1 Elija Herramientas | Opciones del editor y pulse la pestaña Plantillas.
- 2 Utilice el ratón o las flechas *arriba* y *abajo* para seleccionar la plantilla de código deseada.
- 3 Pulse el botón Modificar de la ficha Plantillas para cambiar el nombre o la descripción de la plantilla seleccionada.
- 4 Escriba el nombre y la descripción de la plantilla de código y pulse Aceptar.
- 5 Escriba en el cuadro de texto Código las modificaciones deseadas para el ejemplo de código.
- 6 Añada y modifique las plantillas deseadas, y cuando termine, pulse Aceptar para cerrar el cuadro de diálogo Opciones del editor.

Importante

Cuando se crean y se modifican plantillas, JBuilder no les da el formato correspondiente a las preferencias; las llaves y las sangrías se deben colocar de forma manual.

Si desea más información, pulse el botón Ayuda de la ficha Plantillas del cuadro de diálogo Opciones del editor.

Personalización del editor

El aspecto y el funcionamiento del editor se pueden personalizar de diversos modos. Es posible, por ejemplo, cambiar la fuente y el tamaño de fuente utilizados en el editor, activar y desactivar la numeración de líneas, especificar colores favoritos para los elementos del editor, determinar la posición de los elementos en el panel de estructura, personalizar el comportamiento de las pulsaciones de teclas habituales, etc. Asimismo, es posible especificar el funcionamiento de diversas funciones de CodeInsight, modificar plantillas de código e incluso crear plantillas propias.

El editor se personaliza mediante el cuadro de diálogo Opciones del editor. Para abrir este cuadro de diálogo, seleccione Herramientas | Opciones del editor. Si desea más información sobre las opciones disponibles, pulse el botón Ayuda que aparece en las distintas fichas del cuadro de diálogo Opciones del editor. También puede obtener más información sobre la forma de personalizar el editor si selecciona Ayuda | Entorno de JBuilder y elige el tema “Personalización del editor”.

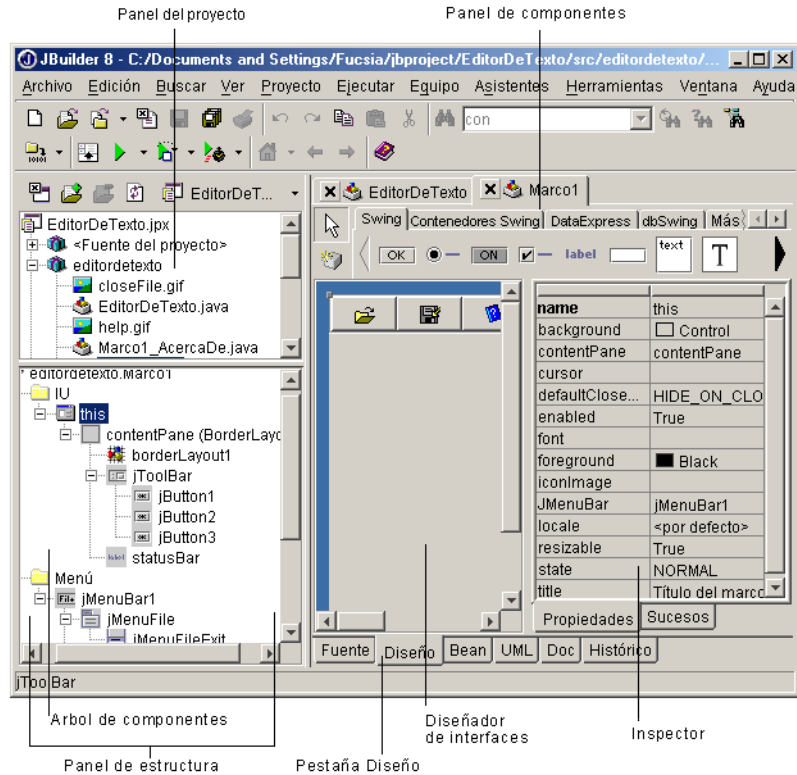
Automatización del desarrollo de código

JBuilder cuenta con gran variedad de herramientas que aceleran el desarrollo de aplicaciones: herramientas de diseño visual que crean rápidamente las interfaces de usuario y asistentes que generan el código automáticamente. Las herramientas de diseño visual incluyen un diseñador de interfaces de usuario, un diseñador de menús y un diseñador de columnas, a los que se tiene acceso desde la pestaña Diseño del panel de contenido. Los asistentes se encuentran disponibles en la Galería de objetos (Archivo | Nuevo) y en el menú Asistentes.

Las herramientas de diseño visual

Las herramientas de diseño visual de JBuilder constan de una paleta de componentes, un inspector, varios diseñadores y un árbol de componentes. Para acceder a las herramientas de diseño es necesario abrir un archivo de fuente en el panel de contenido y hacer clic en la pestaña Diseño.

Figura 6.1 JBuilder en vista diseño



En el centro de la ficha Diseño del panel de contenido hay tres diseñadores: el diseñador de interfaces de usuario, el diseñador de menús y el diseñador de columnas.

El diseñador de interfaces de usuario se utiliza para crear interfaces de usuario que contienen elementos visuales, tales como cuadros de lista y botones. La primera vez que se abre la ficha Diseño después de abrir un archivo en el panel de contenido, JBuilder muestra por defecto el diseñador de interfaces. Los componentes de la interfaz aparecen en la carpeta IU del árbol de componentes.

Si desea información sobre la forma de utilizar el diseñador de interfaces de usuario, consulte “Introducción al diseñador”, en *Diseño de aplicaciones con JBuilder*.

El diseñador de menús se utiliza para crear menús. Los componentes de menú aparecen en la carpeta *Menú* del árbol de componentes. Para acceder al diseñador de menús, haga clic con el botón derecho del ratón en la carpeta *Menú*, o en alguno de sus componentes, y seleccione *Activar el diseñador*.

Si desea información sobre la forma de utilizar el diseñador de menús, consulte “Diseño de menús”, en *Diseño de aplicaciones con JBuilder*.

La Galería de objetos

La Galería de objetos da acceso a asistentes que se pueden utilizar para crear implementaciones básicas de muchos objetos.

Para utilizar un asistente en la Galería de objetos, elija *Archivo | Nuevo* y se abrirá la Galería de objetos. Seleccione un icono de asistente y pulse *Aceptar*. JBuilder abre el asistente asociado, crea la estructura del código en un archivo del tipo adecuado y lo añade al proyecto.

Los asistentes que aparecen atenuados en la Galería de objetos están desactivados. Algunos asistentes no se activan hasta que se crea o se abre un proyecto o un archivo de un tipo determinado. Además, los asistentes pueden estar desactivados si no están disponibles en la edición de JBuilder que se utiliza. Los asistentes varían según la edición de JBuilder.

Importante Para activar algunos asistentes, como los de las fichas *CORBA* y *Enterprise*, es necesario que el servidor de aplicaciones del proyecto se encuentre configurado y seleccionado.

Inicio rápido por medio de los asistentes

Los asistentes de JBuilder permiten ahorrar tiempo, ya que generan el código automáticamente. Se puede acceder a estos asistentes desde varios menús:

- *Archivo | Nuevo*, que abre la Galería de objetos.
- *Menú Asistentes*.
- *Menú Herramientas*.

Proyectos de JBuilder

Todas las tareas que se realiza dentro del IDE de JBuilder se encuadran dentro de un proyecto. Los proyectos de JBuilder son estructuras organizativas que albergan todos los archivos necesarios para ejecutar una tarea, incluidos los directorios en los que residen dichos archivos y todas las vías de acceso, configuraciones y recursos necesarios. El usuario decide el contenido del proyecto.

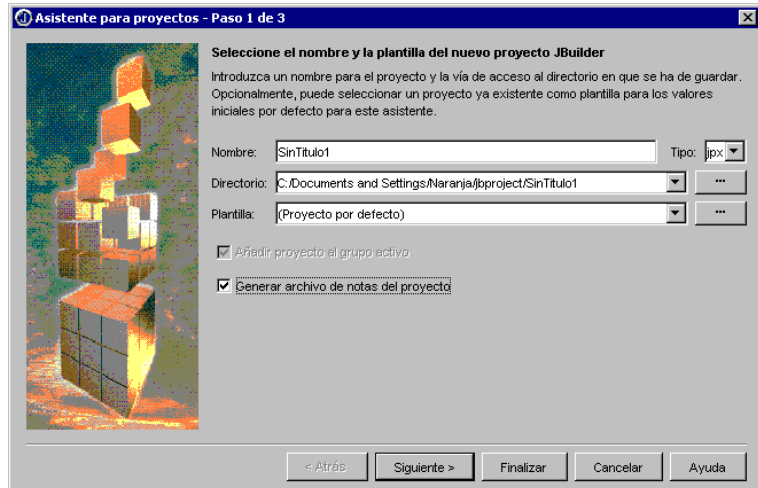
Un proyecto es una unidad lógica, no física. Esto significa que los archivos necesarios para el proyecto pueden residir en cualquier carpeta. La reorganización de la estructura de un proyecto no afecta a los directorios del sistema.

Creación de proyectos

Para crear un proyecto:

- 1 Seleccione Archivo | Nuevo proyecto; o bien, seleccione Archivo | Nuevo para abrir la galería de objetos, pulse la pestaña Proyecto y haga doble clic en el icono Proyecto.

Se abre el Asistente para proyectos.



- 2 En el campo Nombre, escriba el nombre que desee dar al proyecto; por ejemplo, Mi Proyecto.

Puede aceptar el nombre por defecto sugerido por JBuilder o escribir un nombre más descriptivo. JBuilder añade una extensión `.jpx` al nombre del proyecto cuando crea el archivo de proyecto; éste contiene una lista de todos los archivos del proyecto así como sus opciones de configuración. El archivo de proyecto es un documento XML.

Cuando se especifica un nombre de proyecto, éste aparece en el directorio elegido en el campo Directorio, el cual se convierte en el directorio del proyecto. El directorio del proyecto determina cuál es el directorio que contiene el archivo de proyecto. Puede cambiar la ubicación del directorio del proyecto si lo desea; no obstante, por ahora puede utilizar el directorio por defecto.

Si desea conocer otras opciones de la primera ficha del Asistente para proyectos, pulse el botón Ayuda.

- 3 Pulse el botón **Siguiente** para ir a la siguiente ficha del Asistente para proyectos:

La ficha del Asistente para proyectos permite especificar diversas vías de acceso. En la mayoría de los proyectos no es necesario cambiar esas vías de acceso. No obstante, siempre puede hacerlo en el momento en que lo necesite.

El campo JDK indica la versión del JDK que se va a utilizar para el proyecto. JBuilder permite cambiar la versión del JDK mediante el botón JDK.

Vía de salida especifica el lugar donde se almacenan los archivos generados al compilar con JBuilder.

Vía de acceso a copias de seguridad especifica el lugar donde se almacenan las versiones de copia de seguridad de los archivos.

El Directorio de trabajo es el directorio desde el que JBuilder inicia el proyecto.

El campo Vía de acceso por defecto de la ficha Fuente especifica el directorio donde se almacenan todos los archivos de código fuente del proyecto.

Si desea obtener más información sobre estos campos y otros de esta ficha del asistente, pulse el botón Ayuda. En la mayoría de los proyectos sencillos no es necesario hacer cambios.

- 4 Pulse el botón **Finalizar**. (En su lugar, puede pulsar el botón **Siguiente** para ir a la tercera ficha del Asistente para proyectos; sin embargo, esa ficha ofrece funciones más avanzadas que normalmente no son necesarias en los primeros proyectos. Si desea más información sobre el uso de la tercera ficha, ábrala y pulse el botón **Ayuda** del Asistente para proyectos.)

Cuando se pulsa el botón Finalizar, JBuilder crea el proyecto. El archivo de proyecto aparece en el panel del proyecto:

Para obtener más información sobre creación de proyectos, consulte “Creación y gestión de proyectos” en *Creación de aplicaciones con JBuilder*. Para crear proyectos rápidamente a partir de código existente, consulte “Creación de proyectos a partir de archivos existentes” en *Creación de aplicaciones con JBuilder*.

Adición de archivos a un proyecto

En JBuilder SE y JBuilder Enterprise se utiliza la función de detección automática de la vía de acceso a los archivos fuente y se añaden automáticamente las carpetas y los archivos que residen en ese directorio. (Puede desactivar esta función si lo desea; consulte “Recopilación automática de paquetes fuente” en *Creación de aplicaciones con JBuilder* para obtener más información.)

Si utiliza JBuilder Personal, o si tiene desactivada la función de detección automática de la vía de acceso a archivos fuente, añada los archivos y paquetes de los nuevos proyectos mediante los siguientes pasos:

- 1 En la barra de herramientas del panel del proyecto, pulse el botón Añadir Archivos/Paquetes; o bien, haga clic con el botón derecho del ratón en el nodo de archivo de proyecto del panel y, en el menú contextual, seleccione Añadir Archivos/Paquetes.

Se abre el cuadro de diálogo Añadir archivos o paquetes al proyecto.

- 2 Para añadir un archivo, seleccione la ficha Explorador, y para añadir un paquete, la ficha Paquetes.
- 3 Desplácese hasta el archivo o el paquete que desea importar y selecciónelo. Para seleccionar varios nombres de archivo, haga clic en sobre ellos mientras mantiene pulsada la tecla *Mayús* o la tecla *Ctrl*.
- 4 Pulse Aceptar.

El nuevo paquete o archivo aparece como un nodo del panel del proyecto.

Presentación de archivos

Para abrir un archivo de un proyecto de modo que sea posible examinarlo y modificarlo, haga doble clic en el nodo del archivo, en el panel del proyecto. Si un archivo se encuentra dentro de un paquete, empiece por abrir el nodo del paquete. Para ello, haga doble clic en el icono del paquete o pulse el símbolo situado a la izquierda del icono. Cuando pueda ver todos los archivos del paquete, haga doble clic en el que desee abrir.

Los archivos abiertos del proyecto se muestran en el panel de contenido del Visualizador de aplicaciones. Para cada archivo abierto en el proyecto, aparece una pestaña con un nombre de archivo. Puede cerrar rápidamente cualquiera de esos archivos si pulsa el icono X situado en la parte izquierda de la pestaña. Si pulsa la pestaña con el botón derecho del ratón podrá ver otras opciones para cerrar y gestionar el archivo dentro del proyecto.

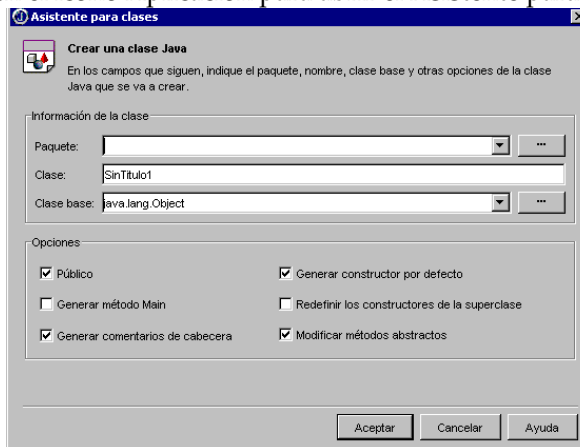
En “Cambio entre archivos” de *Creación de aplicaciones con JBuilder* se explica una forma rápida de cambiar entre los archivos abiertos en el panel de contenido.

Adición de un archivo fuente Java

JBuilder proporciona varias formas de crear archivos Java. Los asistentes de JBuilder suelen crear y añadir varios archivos Java al proyecto. Por ejemplo, el Asistente para clases crea una clase Java, mientras que el Asistente para aplicaciones crea dos archivos Java que constituyen una aplicación Java. A continuación se explica cómo crear y añadir una clase Java a un proyecto y cómo crear un archivo vacío al que añadir cualquier código que se desee.

Para crear una clase Java con el Asistente para clases:

- 1 Seleccione Archivo | Nuevo, pulse la pestaña General y haga doble clic en el icono Aplicación para abrir el Asistente para clases.



- 2 Pulse el botón Ayuda si desea más información sobre el Asistente para clases.
- 3 Rellene los campos del Asistente para clases de la forma apropiada.
- 4 Pulse Aceptar cuando termine.

El asistente crea el archivo de clase especificado y lo coloca en el proyecto. Aparece como un nodo en el panel del proyecto.

Para crear en el proyecto un archivo vacío al que añadir código Java:

- 1 Seleccione Archivo | Archivo nuevo para abrir el cuadro de diálogo Crear archivo o Archivo | Abrir archivo para abrir el cuadro de diálogo Abrir archivo.
- 2 Escriba el nombre del archivo.
- 3 Seleccione el tipo de archivo `.java` en la lista desplegable o incluya esta extensión cuando escriba el nombre.
- 4 Pulse Aceptar.

El nuevo archivo vacío aparece en el panel de contenido, pero no en el panel del proyecto.

- 5 Añada el código que desee al nuevo archivo.
- 6 Para guardar el archivo, seleccione Archivo | Guardar “<nombre de archivo>” o pulse *Ctrl+S*.
- 7 En la barra de herramientas del panel del proyecto, pulse el botón Añadir Archivos/Paquetes; o bien, haga clic con el botón derecho del ratón en el nodo de archivo de proyecto del panel y, en el menú contextual, seleccione Añadir Archivos/Paquetes.

Se abre el cuadro de diálogo Añadir archivos o paquetes al proyecto.

- 8 Abra la ficha Explorador para añadir un archivo.
- 9 Desplácese hasta el archivo recién creado, selecciónelo y pulse Aceptar.

Eliminación y borrado

Los archivos y carpetas de un proyecto se pueden eliminar y borrar. Cuando se eliminan archivos o carpetas se retiran del proyecto, pero no se borran del disco. En cambio, cuando se borran archivos o carpetas, además de eliminarse del proyecto, se borran permanentemente del disco.

Para eliminar archivos, carpetas o paquetes de un proyecto, sin borrarlos del sistema:

- 1 En el panel del proyecto, seleccione los nodos que desee eliminar y, a continuación, en la barra de herramientas del panel, pulse el botón Eliminar del proyecto; *o bien*, haga clic en el nodo con el botón derecho del ratón y seleccione Eliminar del proyecto en el menú contextual.
- 2 Cuando aparezca el cuadro de diálogo Eliminar del proyecto, que solicita la confirmación de la operación, pulse Aceptar.

Los nodos seleccionados desaparecen del panel del proyecto y del propio proyecto. Si se ha seleccionado un nodo de carpeta, los archivos que contiene también se eliminan del proyecto.

Para borrar archivos, carpetas o paquetes del proyecto y del sistema, haga clic con el botón derecho del ratón en el nodo que desee borrar y seleccione Borrar “<nombre del nodo>” en el menú contextual.

Cómo cambiar de nombre proyectos y archivos

Para cambiar el nombre de un proyecto o un archivo:

- 1 Seleccione el proyecto o el archivo en el panel del proyecto.
- 2 Seleccione Proyecto | Cambiar nombre o Archivo | Cambiar nombre, o haga clic con el botón derecho en el panel del proyecto y seleccione Cambiar nombre.
- 3 Escriba el nuevo nombre en el cuadro de diálogo y pulse Aceptar.

También se puede cambiar el nombre de un archivo abierto por medio de su pestaña, en la parte superior del panel de contenido:

- 1 Haga clic con el botón derecho en la pestaña del archivo, situada en la parte superior del panel de contenido.
- 2 Seleccione Cambiar nombre “<nombre del nodo>”.
- 3 Escriba el nuevo nombre en el cuadro de diálogo y pulse Aceptar.

Nota El cambio de nombre de un archivo no modifica su tipo. Si desea cambiar la extensión del archivo, utilice Archivo | Guardar como.

Advertencia Al renombrar proyectos y archivos no se cambian los nombres de paquete y archivo referenciados dentro del código. JBuilder SE y Enterprise ofrecen la función de perfeccionamiento por cambio de nombre, que sustituye el nombre antiguo por el nuevo en todos los lugares donde aparece.

Apertura de proyectos

Para abrir un proyecto:

- 1 Seleccione Archivo | Abrir proyecto.
- 2 Desplácese hasta el directorio que contenga el proyecto que desea abrir. (Si desea información sobre la forma de utilizar el cuadro de diálogo Abrir proyecto, pulse el botón Ayuda.)
- 3 Después de haber seleccionado el directorio del proyecto, colóquese en el panel situado a la derecha del cuadro de diálogo y haga doble clic en

el archivo del proyecto que desea abrir; o bien, seleccione el archivo del proyecto y pulse Aceptar.

Más información sobre los proyectos de JBuilder

Aunque este capítulo presenta suficiente información para empezar a trabajar con los proyectos de JBuilder, también puede consultar “Creación y gestión de proyectos” y “Gestión de las vías de acceso”, en *Creación de aplicaciones con JBuilder*.

El primer capítulo proporciona información más detallada sobre la gestión de proyectos y trata temas y funciones más avanzados. El segundo capítulo explica cómo utiliza JBuilder las vías de acceso para gestionar proyectos, y analiza el uso de las bibliotecas. Es necesario entender bien estos capítulos para saber todo lo que puede hacer JBuilder y cómo lo hace.

Comparación de archivos y versiones

Es una función de JBuilder SE y Enterprise.

JBuilder ofrece muchas posibilidades para comparar archivos, constatar las diferencias entre varios y entre diferentes versiones del mismo archivo, y gestionar, fusionar y restaurar las diferencias. El cuadro de diálogo Comparar archivos y las fichas Histórico proporcionan el acceso a estas funciones.

Glosario de gestión de versiones

En este apartado se utilizan algunos términos especializados:

Término	Definición
<i>revisión, versión</i>	Ambos términos se pueden intercambiar en este contexto. Cada vez que se modifica o se revisa un archivo se hace una nueva versión o revisión de ese archivo.
<i>diferencias</i>	Las diferencias son las áreas de texto que difieren entre dos archivos o entre dos versiones del mismo archivo.
<i>bloque de diferencias</i>	Es una parte que difiere entre dos archivos o dos versiones del mismo archivo. Se aprecian más fácilmente en las vistas de diferencias, en las que los dos archivos o versiones de archivos se fusionan en un archivo aparentemente original y se resaltan las diferencias entre ellos.
	Observe que los mecanismos de gestión de diferencias de JBuilder evalúan los archivos físicamente, no de forma lógica. Solamente comparan el texto de los archivos o las versiones, no su lógica ni su estructura de código.
<i>restaurar, restablecer</i>	Consiste en hacer que la versión anterior de un archivo vuelva a ser la versión actual.

Término	Definición
<i>control de versiones</i>	Es un modo de gestionar las revisiones que guarda un registro de todos los cambios realizados en todos los archivos sometidos a control de versiones. Los sistemas de control de versiones también incluyen normalmente otras funciones de gestión de código, como la ramificación y el etiquetado de versiones.
<i>copia de seguridad</i>	Es una versión del archivo que se guarda en el directorio de copia de seguridad local que se le ha asignado. JBuilder utiliza las copias de seguridad como versiones anteriores del archivo. El número de copias de seguridad que se desea mantener se establece en la ficha Editor de Herramientas Opciones del editor. Si se alcanza el límite, se conservan las últimas versiones y se vuelven a numerar de forma apropiada.
<i>búfer</i>	Es la versión más moderna que se utiliza en JBuilder. Incluye los cambios no guardados.
<i>repositorio</i>	En el control de versiones, es el lugar donde se guardan las copias maestras del archivo y los históricos de revisiones.
<i>área de trabajo</i>	En el control de versiones, son los archivos locales y la estructura de directorio que se cambian directamente.

Comparación de dos archivos

El cuadro de diálogo Comparar archivos de JBuilder permite comparar dos archivos cualesquiera.

Para comparar dos archivos de texto:

- 1 Seleccione Archivo | Comparar archivos.

Aparece el cuadro de diálogo Seleccionar archivo antiguo:

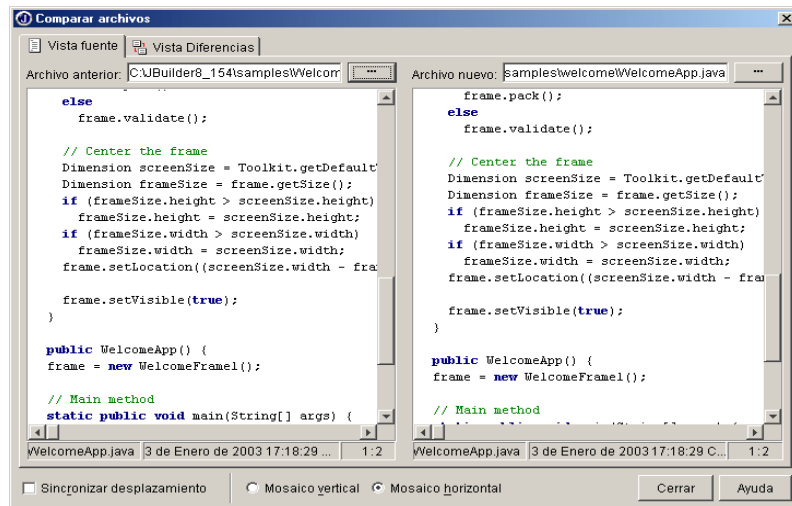


2 Seleccione el primer archivo y pulse Aceptar.

Aparece el cuadro de diálogo Seleccionar archivo nuevo.

3 Seleccione el segundo archivo y pulse Aceptar.

Aparece el cuadro de diálogo Comparar archivos:



La *Vista fuente* muestra un archivo junto al otro. Para dividir la vista en cualquier dirección, pulse el botón de radio apropiado. Para sincronizar el desplazamiento de los dos paneles active la casilla de selección Sincronizar desplazamiento.

La *Vista fuente* admite funciones básicas de proceso de texto. Esto permite llevar a cabo cambios en los dos búfers. Si realiza cambios, aparecen disponibles las opciones Actualizar archivo y Guardar archivo del menú contextual, que se despliega al hacer clic con el botón derecho:



La *Vista diferencias* muestra las diferencias (bloques de diferencias) entre dos archivos. Es posible deshacer bloques de diferencias individuales en el nuevo archivo, de modo que los bloques de código vuelvan a coincidir con el archivo antiguo. El primer archivo seleccionado se muestra como la versión más antigua, y el segundo como la más reciente. Esto significa que si los archivos actual y anterior tienen dos bloques de diferencias en la misma parte del código, JBuilder puede igualarlos sustituyendo el bloque del archivo más moderno por el bloque correspondiente del archivo más antiguo.

Si se modifican los archivos con ayuda del cuadro de diálogo Comparar archivos, cuando se cierra se solicita al usuario que indique si desea guardar los cambios.

La vista Histórico






La vista Histórico permite examinar versiones anteriores de un archivo de cualquier tipo, incluidas las versiones de copias de seguridad, los cambios locales guardados y el búfer del archivo activo. Si el archivo abierto está sometido a control de versiones, en la vista Histórico se ofrecen todos los tipos de revisiones. Si no es así, la vista Histórico permite gestionar las revisiones por medio de las copias de seguridad, los cambios almacenados y el búfer no almacenado.

Como la vista Histórico está en el panel de contenido, sus fichas reflejan el archivo activo actual. Haga doble clic en un archivo, en el panel del proyecto, para activarlo. Pulse la pestaña de un archivo abierto, en el panel de contenido, para activarlo.

Existen cuatro fichas en la vista Histórico. Ofrecen las siguientes funciones:

- La ficha Contenido, disponible en todas las ediciones de JBuilder, muestra versiones actuales y antiguas de los archivos.
- La ficha Diferencias, disponible en JBuilder Professional y Enterprise, muestra las diferencias entre las versiones seleccionadas de un archivo.
- La ficha Información, disponible en JBuilder Enterprise, muestra todas las etiquetas y comentarios del archivo activo.
- La ficha Conflictos en la fusión, disponible en JBuilder Enterprise, muestra los conflictos de fusión notificados por el sistema de control de versiones.

Todas las fichas, excepto Conflictos en la fusión, cuentan con tablas de revisión que enumeran todas las versiones del archivo activo. Las listas de revisión muestran el número de versión del archivo, el tipo y la fecha de revisión, y otros datos. Si desea ordenar las listas de revisión, pulse el encabezado de la columna que desea utilizar como criterio de ordenación. JBuilder utiliza los siguientes iconos en las tablas de revisiones para distinguir los distintos tipos de control de versiones:

Icono	Descripción
	La versión del archivo que se encuentra en el búfer. La versión del búfer incluye los cambios no guardados.
	La versión guardada más antigua del archivo.
	Una versión de copia de seguridad del archivo.
	La versión del archivo extraída del repositorio. Es una función de JBuilder Enterprise.
	Una versión del archivo procedente de un sistema de control de versiones. Es una función de JBuilder Enterprise.

Sugerencia Cuando el cursor se encuentre en la lista de revisiones, desplácese por la lista pulsando la tecla *Intro* o bien utilizando las teclas de desplazamiento. Las fichas Contenido y Diferencias cuentan con un visualizador de código fuente. El visualizador de código fuente muestra el código fuente de la versión del archivo seleccionada en la lista de revisiones. Se puede copiar y pegar del visualizador de código fuente al editor, pero no es posible modificar el código directamente en el visualizador. De esta forma se simplifica la recuperación del trabajo antiguo a la vez que se protege la integridad de las versiones anteriores de los archivos.



Actualizar información de revisiones

El botón Actualizar información de revisiones permite actualizar la vista de la lista de revisiones de modo que incluya los cambios realizados por otros usuarios en el archivo del repositorio. Sólo resulta útil si se utiliza el control de versiones. Este botón está disponible en las tres fichas del Histórico.



Restaurar la revisión anterior

El botón Restaurar la revisión anterior hace que la versión seleccionada vuelva a ser la actual. La versión antigua se convierte en la más reciente. Este botón está en las fichas Contenido e Información. La ficha Información es una función de JBuilder Enterprise.

Nota Al volver a una versión anterior, se pierden todos los cambios no guardados que se encontraban en el búfer del editor. Si se utiliza un sistema de control de versiones, todas las versiones intermedias se almacenan en el repositorio.



Sincronizar desplazamiento

Sincronizar desplazamiento sincroniza el desplazamiento entre los visualizadores de código fuente de las fichas Contenido y Diferencias del Histórico y el editor. Asocia la línea de texto donde está el cursor con la línea correspondiente de la otra vista. Si en esa parte del archivo no hay texto coincidente, asocia los números de línea. Este botón está disponible en la ficha Contenido y Diferencias. La ficha Diferencias es una función de JBuilder SE y Enterprise.

Sincronizar desplazamiento también es una casilla de selección del cuadro de diálogo que aparece cuando se selecciona Archivo | Comparar archivos.

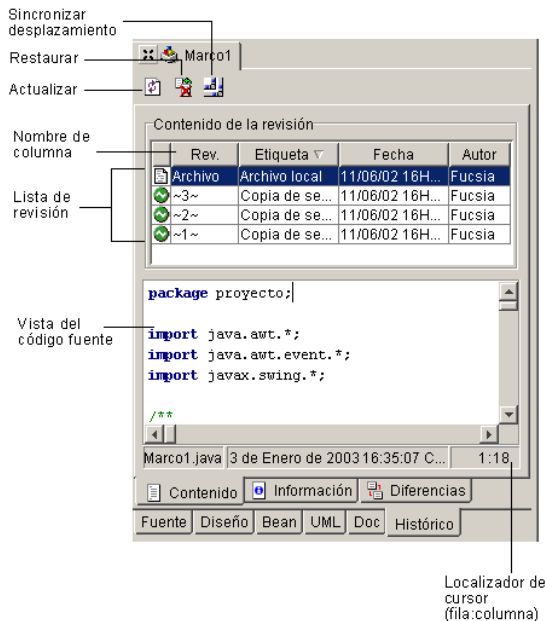


Comparación inteligente

Si la opción Comparación inteligente se encuentra activada, el formato del código y los cambios en los espacios en blanco no se tienen en cuenta cuando se comparan dos versiones de un archivo con JBuilder. Este botón se encuentra en la ficha Diferencias. La ficha Diferencias es una función de JBuilder SE y Enterprise.

Ficha Contenido

La ficha Contenido de la vista del histórico muestra todas las versiones disponibles del archivo activo. La lista de revisiones, en la parte superior, permite ordenar las versiones del archivo por tipo de versión, número de revisión, etiqueta, fecha de cambio y autor. Haga clic en el encabezado que desee utilizar como criterio de ordenación. Si se pulsa de nuevo se invierte el orden (de ascendente a descendente y viceversa). Puede seleccionar una versión en la lista de revisiones y ver su código fuente en el visualizador de código fuente:



Los botones Actualizar información de revisiones, Restaurar la revisión anterior y Sincronizar desplazamiento están activados en esta ficha.

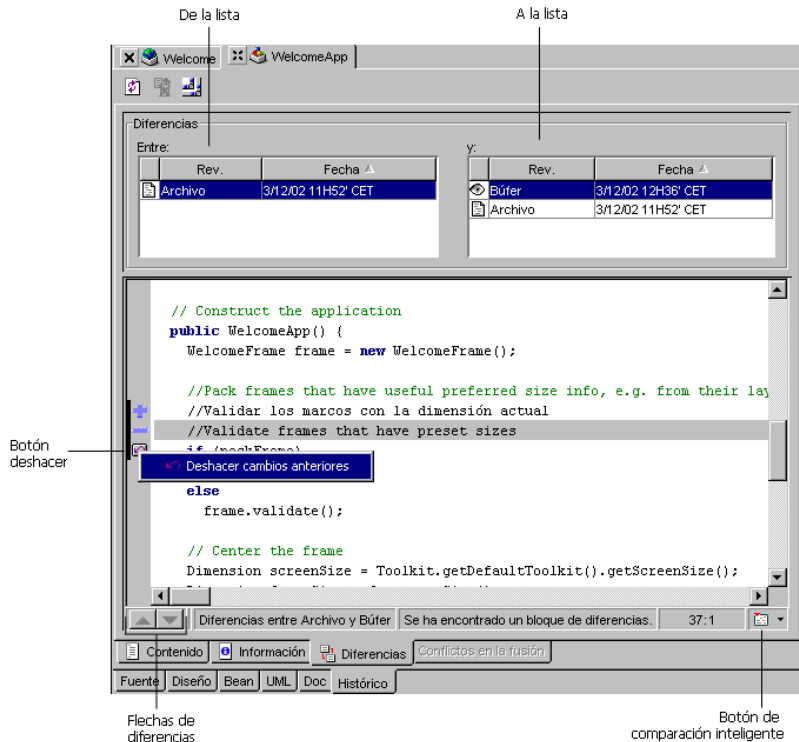
Ficha Diferencias

Es una función de
JBuilder SE y Enterprise.

La ficha Diferencias muestra las diferencias entre dos versiones seleccionadas del archivo activo. Las listas de revisiones de la parte superior permiten ordenar las versiones del archivo por tipo de versión, número de revisión y fecha de cambio. Los botones Actualizar información de revisiones y Sincronizar desplazamiento están activados en esta ficha.

Existen dos listas de revisiones en la ficha Diferencias. Una se denomina Entre, y la otra, En. Seleccione una versión en la lista Entre y otra versión

diferente en la lista En. Las diferencias se muestran en el visualizador de código fuente bajo las listas:



En cada bloque de diferencias, el texto eliminado (con el signo menos) está en color rojo y el texto añadido (con el signo más) está en color amarillo. Si el búfer está seleccionado como la versión Y, pulse el botón **Deshacer** si desea deshacer la adición.

Utilice las flechas de diferencias para desplazarse entre los bloques de diferencias. Si existen muchos bloques de diferencias debidos a cambios en el formato del código, utilice **Comparación inteligente** para omitir esos cambios (tales como ubicación y número de líneas en blanco, sangrado diferente o uso de espacios).

Sugerencia Cuando el cursor se encuentre en el panel de la vista de código fuente de la ficha **Diferencias**, puede pulsar **Mayús+F10** para abrir un menú contextual que contiene opciones para activar y desactivar **Comparación inteligente**, así como desplazarse a bloques de diferencias anteriores y posteriores.

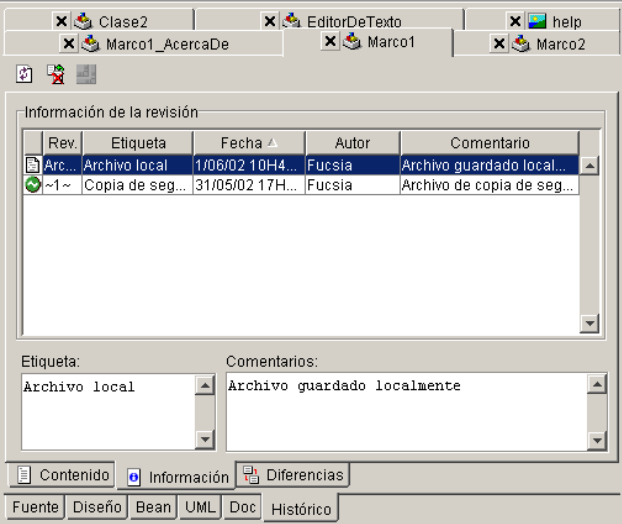
Nota Los usuarios del control de versiones pueden visualizar todos los cambios realizados en el repositorio desde la extracción original. Seleccione la versión original del repositorio en el área Entre y la versión del archivo con el número más alto de revisiones del área Y. JBuilder actualiza la versión más reciente con los cambios del repositorio.

Ficha Información

Es una función de JBuilder Enterprise.

La ficha Información de la vista Histórico muestra el texto completo de las etiquetas y comentarios de cualquier versión seleccionada del archivo activo. La lista de revisiones, en la parte superior, permite ordenar las versiones del archivo por tipo de versión, número de revisión, etiqueta, fecha de cambio, autor y comentario. Los botones Actualizar información de revisiones y Restaurar la revisión anterior están activados en esta ficha.

Para utilizar la ficha Información, seleccione una versión en la lista de revisiones. Su etiqueta (si es que tiene) y cualquier comentario introducido para describir la revisión se muestran en la parte inferior de la ficha:



La ficha Información es una herramienta de investigación. Permite desplazarse por los comentarios de todas las revisiones de un archivo para obtener una vista general de todos los cambios llevados a cabo.

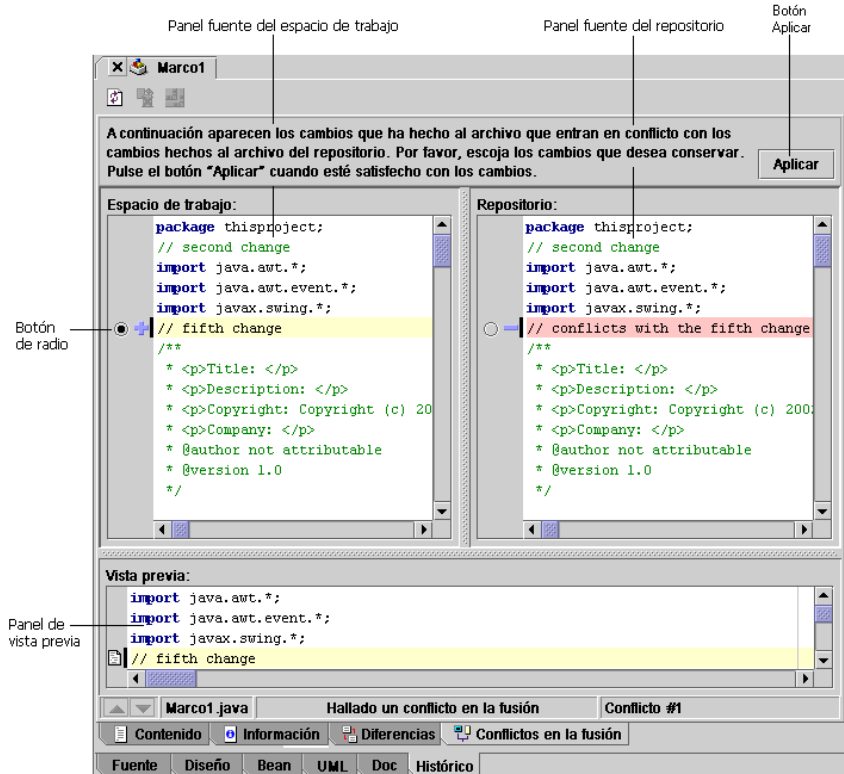
Si desea información adicional sobre estas funciones, pulse la tecla *F1* desde cualquiera de las fichas del histórico. De esta forma aparece la ayuda contextual del panel de contenido. Seleccione Histórico en la tabla de la ayuda contextual.

Ficha Conflictos en la fusión

Es una función de JBuilder SE y Enterprise.

La ficha Conflictos en la fusión permite resolver automáticamente los conflictos de fusión que se pueden producir al trabajar en la integración entre JBuilder y sistemas como Concurrent Versioning System (CVS) o Visual SourceSafe (VSS). La ficha Conflictos en la fusión sólo aparece activada cuando se han detectado conflictos.

Cuando se produce un conflicto de fusión, la ficha Conflictos en la fusión muestra el código fuente del espacio de trabajo junto al código fuente del repositorio, con los bloques de código o texto en conflicto resaltados. Los botones circulares contiguos a los bloques resaltados de código conflictivo permiten seleccionar fácilmente el bloque de código que se desea mantener. El panel de vista preliminar, en la parte inferior de la ficha Conflictos en la fusión, muestra el aspecto que tendrá el archivo del espacio de trabajo cuando se le hayan aplicado los cambios.



Las líneas de un conflicto que se han seleccionado para mantenerse se marcan en amarillo, y además presentan el signo más (+) en el margen. Los cambios que se descartan se marcan en rojo y tienen el signo menos (-) en el margen. Si existen muchos conflictos, se pueden pulsar las flechas de desplazamiento que se encuentran en la esquina izquierda inferior de la

ficha para desplazarse hacia delante y atrás entre los pares de diferencias. El panel de vista preliminar, en la parte inferior de la ficha Conflictos en la fusión, muestra el aspecto que tendrá el archivo original cuando se le hayan aplicado los cambios. Por defecto, los cambios en el espacio de trabajo se seleccionan cuando se desea mantenerlos.

Después de haber seleccionado los cambios que desee conservar, pulse Aplicar para almacenarlos en el búfer del editor. De esta forma se actualiza el archivo en el que se está trabajando. La aplicación los cambios no significa que se guarde el archivo o se incorporen (confirmen) los cambios en el repositorio. Hasta que se confirman los cambios, se puede utilizar el comando Deshacer (Edición | Deshacer o *Ctrl+Z*) para restablecer los conflictos. El comando Deshacer no se encuentra disponible desde el panel del histórico, por lo que es necesario cambiar a otra vista.

La ficha Conflictos en la fusión sólo aparece activada cuando se producen conflictos de fusión al utilizar la integración entre JBuilder y sistemas como Concurrent Versioning System (CVS) o Visual SourceSafe (VSS). Si desea más información sobre la forma de resolver conflictos de fusión con CVS o VSS, consulte “Control de versiones de proyectos ya creados mediante CVS” o “Control de versiones de proyectos ya creados mediante VSS”, en *Desarrollo en equipo con JBuilder*.

Tutorial: Creación de un applet

Este tutorial indica los pasos necesarios para crear un applet AWT en el entorno integrado de desarrollo (IDE) de JBuilder. Para obtener más información sobre el IDE y sus componentes, consulte “El entorno de JBuilder”, disponible en el menú Ayuda.

El tutorial muestra cómo:

- Crear un proyecto para la aplicación.
- Utilizar el Asistente para applets para crear un applet AWT.
- Compilar y ejecutar la aplicación.
- Personalizar la interfaz del applet.
- Añadir componentes AWT, tales como Choice, Label y Button.
- Modificar el código fuente.
- Distribuir el applet.
- Modificar el archivo HTML.
- Ejecutar el applet distribuido desde la línea de comandos.
- Comprobar el applet.

El tutorial completo se suministra como un ejemplo. Para ver los archivos de código fuente o ejecutar la aplicación, abra el archivo `FirstApplet.jpx` del directorio `<jbuilder>/samples/Tutorials/FirstApplet`. Los usuarios con acceso de sólo lectura a los ejemplos de JBuilder deben copiar el directorio `samples` en un directorio para el que tengan permiso de lectura y escritura.

Importante Antes de comenzar este tutorial, lea la descripción general, donde se tratan importantes cuestiones sobre los applets.

Consulte el apartado Tutoriales en las Sugerencias de JBuilder si desea información sobre cómo ver e imprimir tutoriales. El apartado Opciones de accesibilidad de Sugerencias de JBuilder contiene sugerencias sobre la utilización de las funciones de JBuilder para mejorar la facilidad de uso de JBuilder para personas con discapacidades.

Para obtener información sobre las convenciones utilizadas en este tutorial y en otra documentación de JBuilder, consulte [“Convenciones de la documentación” en la página 2-4](#).

Aspectos generales

Durante el diseño de applets, conviene recordar que la compatibilidad con Java que ofrecen los navegadores es limitada. En el momento de la redacción de la presente documentación, Microsoft Internet Explorer y Netscape Navigator son compatibles con el JDK 1.1.5. Los navegadores actuales no son compatibles con los componentes Swing presentes en el JDK 1.1.7, aunque podrían integrarlos en el futuro. Si crea un applet con una versión posterior del JDK, debe utilizar sólo componentes compatibles con los navegadores. Por ejemplo, si desarrolla los applets sólo con componentes AWT, en la mayoría de los casos se ejecutarán sin problemas. Los componentes AWT pueden haber cambiado (por ejemplo, el JDK 1.1.x puede ofrecer funciones ligeramente distintas de las del JDK 1.3) y es posible que tenga que modificar el código del applet para que se adapte a sus necesidades. Los errores del applet se pueden resolver en el navegador mediante el examen de los mensajes de error de la Consola Java. La forma más segura de diseñar un applet consiste en utilizar los componentes AWT, así como el JDK compatible con el navegador.

Nota Los navegadores compatibles con JDK 1.1 son, entre otros, Netscape 4.06 e Internet Explorer 4.01 y versiones posteriores de ambos. La compatibilidad con la versión del JDK varía según la plataforma.

Otra posibilidad consiste en diseñar los applets utilizando el JDK y los componentes Swing actuales y facilitar a los usuarios el Plug-in de Java. Esto sólo es factible en un entorno controlado, como la intranet de una empresa. Los navegadores compatibles con el Plug-in de Java, que dota a la aplicación del Java 2 SDK 1.3 Runtime Environment (JRE), pueden ejecutar applets basados en el JDK 1.3. Existen pasos adicionales en el desarrollo de applets que se ejecutan con el Plug-in Java. Si desea más información, consulte la página web del Plug-in Java en <http://www.javasoft.com/products/plugin/index.html>.

Aunque los navegadores más recientes sólo aceptan el JDK 1.1.5, en este tutorial se va a utilizar el JDK 1.3 y sólo se van a emplear los componentes AWT para el diseño, sin recurrir a las nuevas funciones del JDK 1.3. Sin embargo, el applet se ejecutará sin problemas porque los componentes que se van a utilizar se han seleccionado cuidadosamente. Los

navegadores antiguos que utilizan el JDK 1.02 no pueden ejecutar este applet, porque la gestión de sucesos cambió en el JDK 1.1. Si va a utilizar el applet en navegadores antiguos, debe emplear la gestión de sucesos del JDK 1.02.

Importante Si desea más información sobre la ejecución de los applets de los JDK 1.1.x y 1.2/1.3 en JBuilder, consulte “Ejecución de applets” en *Guía del desarrollador de aplicaciones web*.

El applet Buenas noches

El applet “Buenas noches” que se crea en este tutorial contiene una lista desplegable de opciones de idioma. Cuando se selecciona un idioma, como el alemán, el panel situado debajo de la selección muestra la traducción al alemán de “Buenas noches”: “Guten Abend”.

Cuando acabe este tutorial, si ejecuta el applet en el visualizador de applets de JBuilder, verá que tiene un aspecto parecido a éste:



Para obtener información más detallada acerca de los applets y su distribución, consulte “Los applets” en *Guía del desarrollador de aplicaciones web* y “Distribución de programas en Java” en *Creación de aplicaciones con JBuilder*.

Paso 1: Creación de un proyecto

Antes de comenzar este tutorial, lea “Aspectos generales” en la página 9-2, que trata cuestiones importantes sobre las applets, como la compatibilidad con navegadores, las versiones del JDK y los componentes de las applets.

Antes de crear el applet necesita un proyecto en el que guardarlo. El Asistente para proyectos lo crea automáticamente.

1 Elija Archivo | Nuevo proyecto para iniciar el Asistente para proyectos.

- 2 En el primer paso del Asistente para proyectos, realice los siguientes cambios en los campos correspondientes:

- a Escriba `FirstApplet` en el campo Nombre.

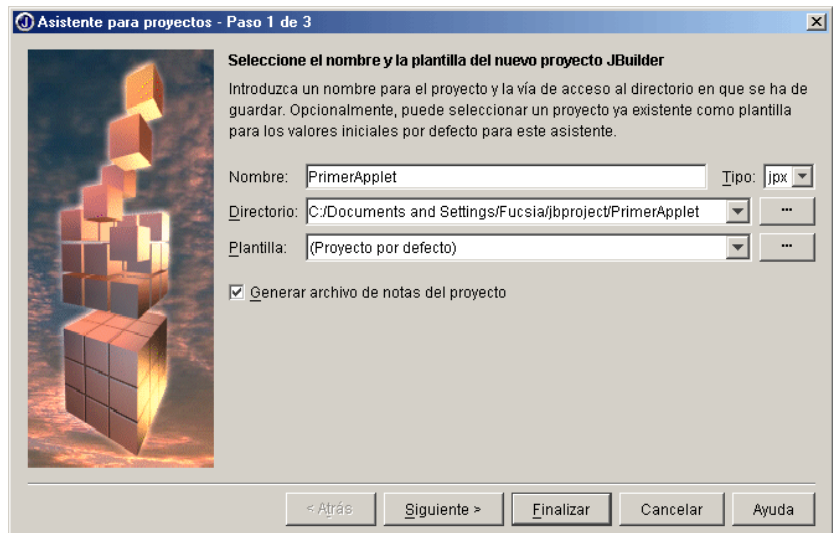
Nota

Cuando se escribe el nombre del proyecto en el campo Nombre, se escribe el mismo nombre en el campo Directorio. Por defecto, JBuilder utiliza este nombre de proyecto para crear los nombres del directorio del proyecto y el paquete de las clases. Por defecto, los proyectos de JBuilder se guardan en el directorio `<inicio>/jbproject`. Consulte [“Convenciones de la documentación” en la página 2-4](#). Si desea más detalles sobre los proyectos, consulte “Creación y gestión de proyectos” en *Creación de aplicaciones con JBuilder*.

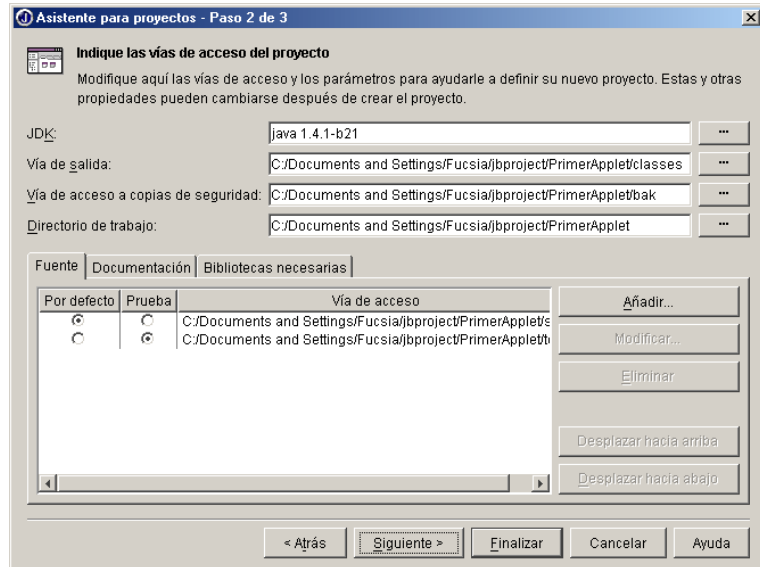
- b Seleccione `jspx` en el menú desplegable Tipo.
- c Seleccione la opción Generar archivo de notas del proyecto.

Cuando se activa esta opción, el Asistente para proyectos crea un archivo HTML para las notas del proyecto y lo añade.

- 3 Acepte todas las demás opciones por defecto del Paso 1.



4 Pulse Siguiente para ir al Paso 2 del Asistente para proyectos.



5 Acepte las vías de acceso por defecto del Paso 2. Observe dónde se guardan los archivos de clase compilados, los archivos de proyecto y los archivos de código fuente. Si desea más detalles sobre las vías de acceso y cómo guarda JBuilder los archivos, consulte “Gestión de las vías de acceso” en *Creación de aplicaciones con JBuilder*.

Sugerencia

Si prefiere crear un applet con una versión anterior del JDK, modifique la versión del JDK en este paso. Aunque JBuilder Personal no permite cambiar de JDK, se puede modificar el JDK empleado en el cuadro de diálogo Configurar JDK (Herramientas | Configurar JDK). En el caso del JDK 1.1.x también es necesario descargar la versión del JFC específica del JDK 1.1.x.

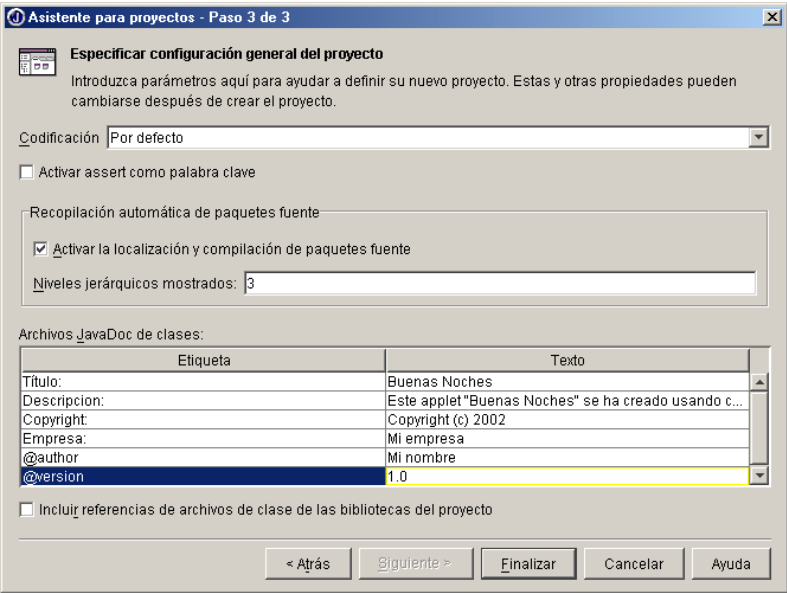
Consulte

- Si desea más información sobre la modificación o la edición de JDK, consulte “Configuración de las propiedades de proyecto” en *Creación de aplicaciones con JBuilder*.
- “Ejecución de applets” en *Guía del desarrollador de aplicaciones web* para obtener más información sobre la ejecución de applets en JBuilder
- “Cómo construye JBuilder las vías de acceso” en *Creación de aplicaciones con JBuilder*.
- “Localización de los archivos” en *Creación de aplicaciones con JBuilder*.

6 Haga clic en Siguiente para avanzar al Paso 3 del asistente.

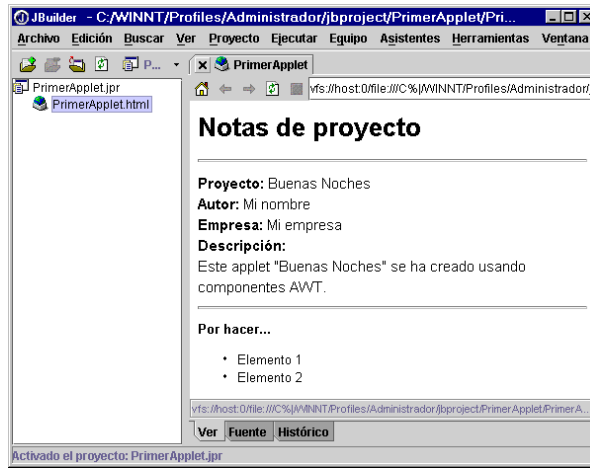
- 7 Escriba lo siguiente en los campos apropiados del Paso 3:
- a Acepte las opciones por defecto para Codificación y Recopilación automática de paquetes fuente. La recopilación automática de paquetes fuente es una función de JBuilder SE y Enterprise.
 - b Escriba Buenas noches en el campo Título de los campos de clase Javadoc.
 - c Escriba su nombre, el nombre de la empresa y una descripción de la aplicación en los campos optativos correspondientes.

Nota La información de los campos de clase Javadoc aparece en el archivo del proyecto HTML y en forma de comentarios de cabecera optativos en el código fuente.



- 8 Pulse el botón Finalizar.
- El asistente crea un archivo de proyecto y un archivo de notas de proyecto, PrimerApplet.jpr y PrimerApplet.html, que aparecen en el panel del proyecto del Visualizador de aplicaciones.

- 9 Haga doble clic en el archivo HTML para ver las notas del proyecto en el panel de contenido.



Paso 2: Generación de los archivos de código fuente

El Asistente para applets crea un archivo .java y un archivo HTML del applet y los coloca en el proyecto que se acaba de crear con el Asistente para proyectos.

Para generar los archivos fuente del applet, siga estas instrucciones:

- 1 Elija Archivo | Nuevo y seleccione la pestaña Web de la galería de objetos.
- 2 Haga doble clic en el icono Applet para iniciar el Asistente para applets.
- 3 Acepte el nombre del paquete por defecto, `primerapplet`, en el Paso 1. Por defecto, el asistente asigna al paquete el nombre del proyecto, `PrimerApplet.jpr`.
- 4 Escriba `GoodEveningApplet` en el campo Clase.

Se trata de un nombre de clase de Java que distingue entre mayúsculas y minúsculas.

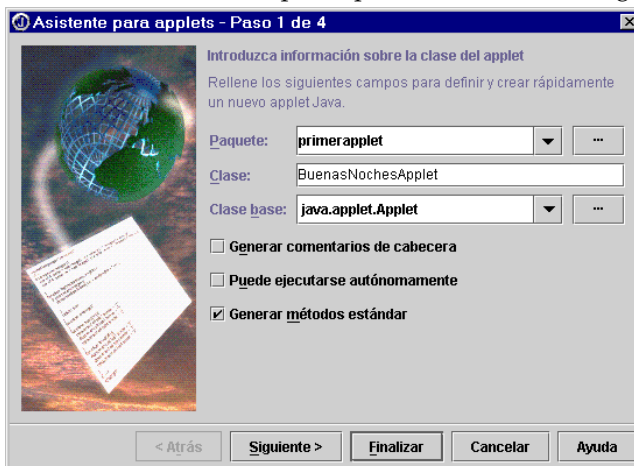
Nota El nombre completo de la clase (nombre del paquete + nombre de la clase) es `primerapplet.BuenasNochesApplet.class`

- 5 Acepte la superclase por defecto, `java.applet.Applet`.

Advertencia Si crea el applet con `javax.swing.JApplet`, no se ejecutará en la mayoría de los navegadores web. En el momento de la redacción de este manual, los navegadores no son compatibles con Swing. Consulte ["Aspectos generales"](#) en la página 9-2 para obtener más información.

- 6 Active la opción Generar métodos estándar:

El Paso 1 del Asistente para aplicaciones ofrece el siguiente aspecto:



7 Pulse Siguiente para pasar al Paso 2.

En este paso se pueden añadir parámetros al applet. El Asistente añade las etiquetas `<param>` dentro de las etiquetas `<applet>` en el archivo HTML del applet, y también inserta código para gestionar los parámetros del código fuente. Los parámetros de los applets, equivalentes a los argumentos de línea de comandos de las aplicaciones, permiten personalizar el applet. No añada ningún parámetro a este tutorial.

Consulte

- “Defining and using parameters” en <http://www.java.sun.com/docs/books/tutorial/applet/.appletonly/param.html>

8 Pulse Siguiente para avanzar al Paso 3 del Asistente para applets.

9 Realice los siguientes cambios en el Paso 3:

- a** Acepte la opción por defecto, Generar página HTML.

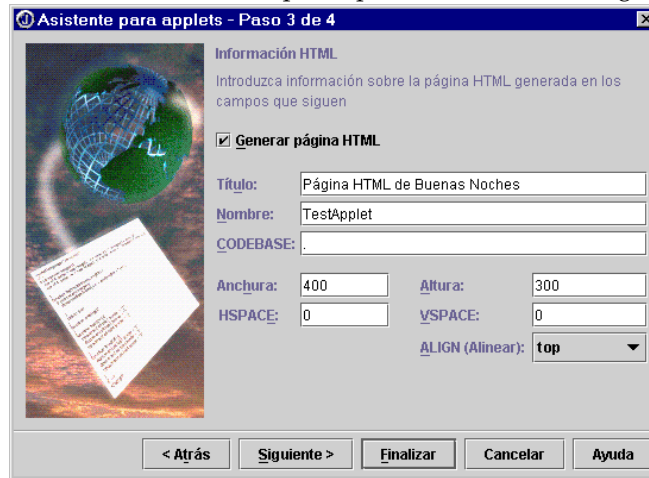
Cuando se selecciona esta opción se crea el archivo HTML que llama al applet. Este archivo HTML contiene la etiqueta `<applet>` y diversos atributos.

- b** Escriba Página HTML de Buenas noches en el campo Título.

El título aparece en la ventana del navegador cuando se ejecuta el applet.

- c** Acepte los valores por defecto de los demás atributos.

El Paso 3 del Asistente para aplicaciones ofrece el siguiente aspecto:



La etiqueta `<applet>` del archivo HTML puede incluir estos atributos:

codebase	Este atributo optativo especifica la vía de acceso correspondiente al archivo HTML del applet en el que el navegador busca los archivos de clase necesarios. El valor “.” especifica el mismo directorio como el archivo HTML que ejecuta el applet. El atributo codebase es necesario cuando los archivos de clase se encuentran en un directorio distinto al del archivo HTML.
código	Este atributo necesario, que inserta automáticamente el Asistente para applets de JBuilder, es el nombre completo de la clase (nombre de paquete + nombre de clase) del applet que contiene el método <code>init()</code> . Lo verá en el archivo HTML al generarlo. En este ejemplo, el nombre completo de la clase es <code>primerapplet.AppletBuenasNoches.class</code> .
archive	Este atributo optativo, que no se incluye en el código generado por el Asistente para applets, es necesario cuando se distribuye el applet en un archivo JAR, ZIP o CAB. Los archivos comprimidos deben estar en el directorio especificado por codebase.
name	Este atributo opcional indica el nombre del applet.
width/height	Estos atributos necesarios determinan el ancho y la altura del área de representación del applet en píxeles.

hspace/vspace	Estos atributos opcionales determinan el tamaño adicional horizontal (márgenes izquierdo y derecho) y vertical (márgenes superior e inferior) que rodea al applet en píxeles.
align	Este atributo opcional determina la alineación del applet en la página HTML.

Importante Los valores de `codebase`, `code`, `archive` y `name` deben estar entre comillas y distinguen entre mayúsculas y minúsculas.

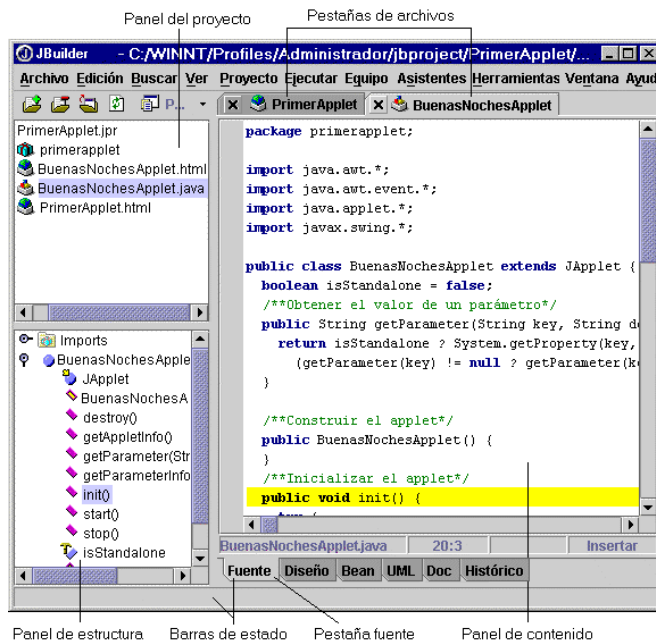
Consulte

- “Atributos de las etiquetas `<applet>`” en *Guía del desarrollador de aplicaciones web*.

10 Pulse Finalizar para cerrar el Asistente para applets.

Observe que se crean dos archivos, `BuenasNochesApplet.java` y `BuenasNochesApplet.html`, y se añaden al proyecto. `BuenasNochesApplet.java` se abre en el panel de contenido.

Nota En las ediciones de JBuilder SE y Enterprise también aparece automáticamente un nodo de paquete de código fuente llamado `HolaTodos` si se ha elegido la opción Activar la localización y compilación de paquetes de código fuente de la ficha General del cuadro de diálogo Propiedades de proyecto (Proyecto | Propiedades de proyecto).



A continuación, observe el código fuente de los dos archivos.

1 Mire el archivo `BuenasNochesApplet.java` y observe lo siguiente:

- Contiene el método `init()`. El archivo HTML debe llamar a la clase que contiene el método `init()` para que se ejecute el applet.
- El nombre de paquete `primerapplet` es la primera línea del código. El archivo de clase se guarda en un directorio `primerapplet`, según las convenciones de Java.
- Las sentencias de importación importan paquetes AWT, no Swing:

```
import java.awt.*;
import java.awt.event.*;
import java.applet.*;
```

2 Haga doble clic en `BuenasNochesApplet.html`, en el panel del proyecto, y seleccione la pestaña Fuente en la parte inferior del panel de contenido para ver el código fuente.

Observe que el asistente ha insertado el valor de `code`, `primerapplet.BuenasNochesApplet.class`.

3 Seleccione Archivo | Guardar todo para guardar los archivos de código fuente y el archivo de proyecto.

Nota Por defecto, JBuilder guarda los archivos fuente en:

```
/<inicio>/jbproject/PrimerApplet/src/primerapplet
```

El archivo HTML del applet se guarda en el directorio `classes`:

```
/<inicio>/jbproject/PrimerApplet/classes/
```

Tras la compilación, los archivos de clase se guardan en la vía de acceso de archivos generados:

```
/<inicio>/jbproject/PrimerApplet/classes/primerapplet
```

JBuilder siempre respeta la jerarquía de paquetes cuando guarda archivos. En este ejemplo, los archivos de código fuente y de clase se guardan dentro de un directorio `primerapplet`, en las vías de acceso a fuente y a archivos generados, respectivamente, para reflejar la estructura de paquete `primerapplet`. El Asistente para proyectos configura estas vías de acceso, que se pueden ver en el cuadro de diálogo Propiedades de proyecto (Proyecto | Propiedades de proyecto). En este tutorial se han aceptado las vías de acceso por defecto de JBuilder en el Paso 2 del asistente para proyectos.

Paso 3: Compilación y ejecución del applet

Ahora, compile y ejecute el applet. La compilación es el proceso de ejecutar el compilador Java. El compilador, que convierte el código fuente en bytecode de Java, genera archivos `.class`.

Importante Si desea más información sobre la ejecución de los applets de los JDK 1.1.x y 1.2/1.3 en JBuilder, consulte “Ejecución de applets” en *Guía del desarrollador de aplicaciones web*.”.



1 Elija Ejecutar | Ejecutar el proyecto, o haga clic en el botón Ejecutar para compilar y ejecutar el applet. El menú Ejecutar y el botón Ejecutar ejecutan la aplicación en el appletviewer de JBuilder, AppletTestbed.

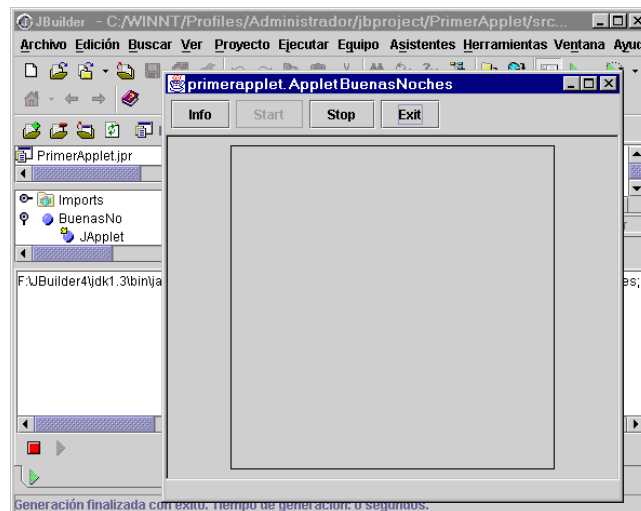
Sugerencia

También puede hacer doble clic en `AppletBuenasNoches.html` en el panel de proyectos y seleccionar Ejecutar. De esta forma se ejecuta el applet en la herramienta de línea de comandos `appletviewer` de Sun.

La ventana está vacía, porque el applet no cuenta aún con una interfaz de usuario.

Cuando se ejecuta el applet, en la parte inferior del Visualizador de aplicaciones aparece el panel de mensajes, donde se muestran todos los errores surgidos durante la compilación. Corrija todos los errores y ejecute el applet de nuevo.

Se muestra el applet, que debe tener un aspecto parecido al que aparece a continuación en AppletTestbed de JBuilder:



Es posible modificar los parámetros de ejecución del applet en la ficha Ejecutar del cuadro de diálogo Propiedades de proyecto. Para acceder a él, abra el cuadro de diálogo Propiedades de proyecto (Proyecto | Propiedades de proyecto), abra la ficha Ejecutar, seleccione `BuenasNochesApplet` y pulse Modificar. La configuración de la pestaña Applet controla el comportamiento del menú Ejecutar y el icono Ejecutar de la barra de herramientas. Elija Clase principal para ejecutar el applet en AppletTestbed de JBuilder. Elija HTML para ejecutar el applet en el `appletviewer` de Sun. Cuando el applet se crea con el asistente, la opción activada por defecto es Clase principal.

Importante

Las applets se ejecutan desde el archivo HTML, que llama a la clase que contiene el método `init()`, no desde el archivo `.java`. Si se intenta ejecutar el archivo `.java` aparece un mensaje de error (a menos que se haya seleccionado la opción Puede ejecutarse autónomamente en el Paso 1 del Asistente para applets):

```
java.lang.NoSuchMethodError: principal
Exception in thread "main"
```

- 2 Pulse Salir en el applet “Buenas noches” para cerrarlo.
- 3 Haga clic con el botón derecho del ratón en la pestaña `BuenasNochesApplet` del panel de mensajes y seleccione Eliminar pestaña “BuenasNochesApplet” para cerrar los mensajes de ejecución.

Paso 4: Personalización de la interfaz de usuario del applet

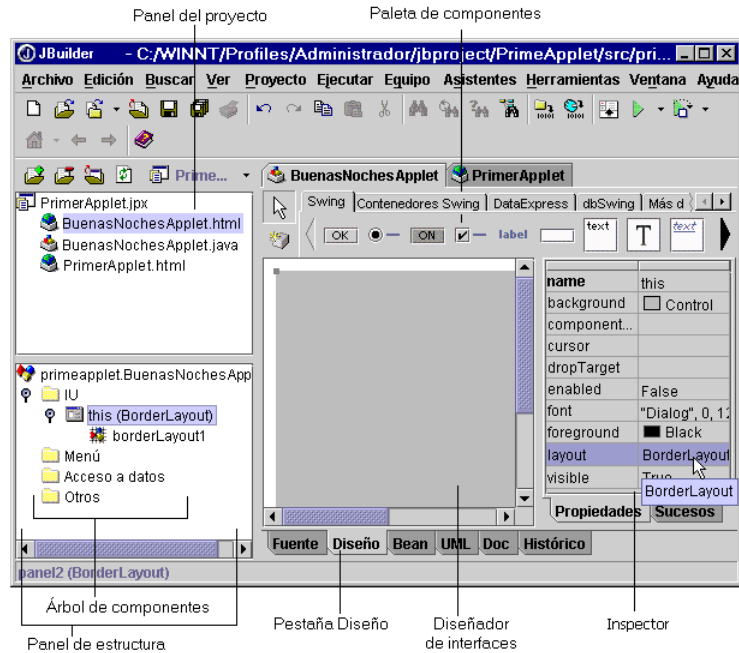
Ahora que el Asistente para applets ha generado el shell del applet, se va a personalizar con varios componentes.



- 1 Seleccione la pestaña del archivo `BuenasNochesApplet.java`, en la parte superior del panel de contenido.
- 2 Seleccione la pestaña Diseño de la parte inferior del panel de contenido para cambiar al modo de diseño.

El diseñador de interfaces de usuario aparece en el panel de contenido, con la paleta de componentes arriba y el Inspector a la derecha. La paleta de componentes se utiliza para añadir componentes a la interfaz, y el Inspector, para modificar propiedades y añadir sucesos al código.

Ahora, el panel de estructura contiene un árbol de componentes con carpetas como IU, Menú y Otros.



- 3 Cambie el diseño del elemento `this` a `BorderLayout` en el Inspector.
 - a Seleccione el nodo `this` en el árbol de componentes del panel de estructura.
 - b Haga clic a la derecha de la propiedad `layout` de la ficha Propiedades del Inspector.
 - c Seleccione `BorderLayout` de la lista desplegable de diseños.

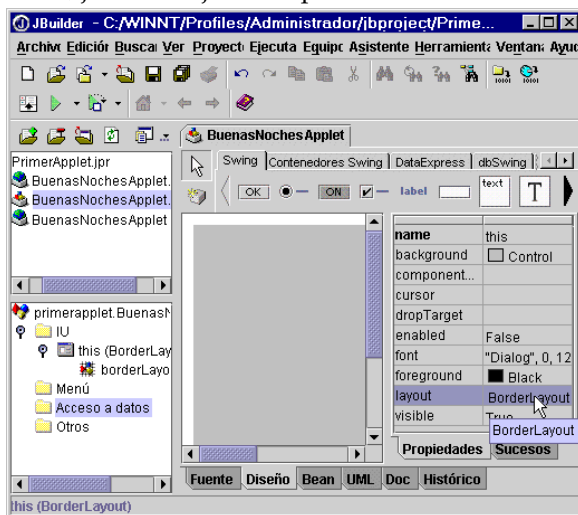
`BorderLayout` organiza los componentes de los contenedores en áreas denominadas North (Norte), South (Sur), East (Este), West (Oeste) y Center (Centro). Utilice `BorderLayout` cuando quiera llevar los componentes a alguno de los bordes de un contenedor o rellenar el centro del contenedor con un componente. Éste es también el diseño que se debe utilizar si se desea que un componente llene por completo el contenedor.

Advertencia

Si se elige `XYLayout` (un diseño personalizado de Borland) en la lista desplegable del Inspector, JBuilder añade esta sentencia de importación al código fuente: `com.borland.jbcl.layout.*`. Esta sentencia de importación *no* se elimina cuando se cambia a un diseño más portable antes de la distribución. El applet distribuido no se ejecuta si no se elimina esta sentencia de importación, ya que busca las clases `jbcl` que

debe importar. Esta sentencia se debe borrar de forma manual antes de distribuir el applet.

El diseñador de interfaces de usuario de JBuilder utiliza un gestor de diseño por defecto para cada contenedor. Normalmente se trata del diseño del contenedor AWT superior. En el AWT de Java, todos los paneles utilizan `FlowLayout` por defecto. Para ver el gestor de diseño del panel, pulse el icono de ampliación del árbol de componentes para ampliar la selección. El gestor de diseños aparece como un elemento del árbol, justo debajo del supercontenedor.



- 4 Pulse la ficha AWT de la paleta de componentes, en la parte superior del diseñador de interfaces.

Puede que la pestaña AWT se encuentre fuera de la pantalla; en este caso es necesario desplazarse a la derecha de la paleta de componentes. Los componentes AWT, a diferencia de los componentes Swing, son compatibles con la mayoría de los navegadores.

Advertencia

El diseñador de interfaces de usuario siempre se abre con los componentes de Swing en la primera ficha de la paleta de componentes. Si se seleccionan estos componentes de Swing por error, el applet no se ejecuta en los navegadores web. Cerciérese de que cuando esté colocando componentes lo haga desde la ficha AWT de la paleta.



- 5 Ahora se deben añadir dos paneles al panel `this`. El panel superior contendrá una lista desplegable de idiomas en la que se podrá seleccionar un elemento, y una etiqueta para identificar la lista. El panel inferior contendrá la frase “Buenas noches” en varios idiomas.

a Pulse *Mayús* mientras hace clic en el componente `panel`, `java.awt.Panel`, en la paleta de componentes.

Mayús+clic permite colocar el mismo componente varias veces sin necesidad de seleccionarlo cada vez en la paleta de componentes. Esto permite añadir varios paneles a `this`.

Sugerencia

Si se coloca el cursor sobre un componente aparecen su nombre en una etiqueta de ayuda inmediata.

b Haga clic dos veces en `this`, en el árbol de componentes del panel de estructura, para añadir dos paneles. `panel 1` y `panel 2` se añaden bajo `this` en el árbol de componentes.



c Haga clic sobre la herramienta de Selección a la izquierda de la paleta de componentes para desactivar la selección del componente.

d Active la propiedad `constraints` de los paneles en el Inspector. La del panel superior debe ser `North`; la del inferior, `Center`. Para comprobar la propiedad `constraints`:

1 Seleccione un panel en el árbol de componentes o en el diseñador de interfaces de usuario. Cuando un componente está seleccionado en el diseñador aparecen tiradores en sus esquinas.

2 Compruebe en el Inspector que la propiedad `constraints` tiene el valor `North` para el panel superior y `Center` para el inferior. Si no es así, haga clic en la columna situada a la derecha de la propiedad `constraints` y seleccione la restricción adecuada en la lista desplegable.

e Asigne al panel superior el nombre `superior`.

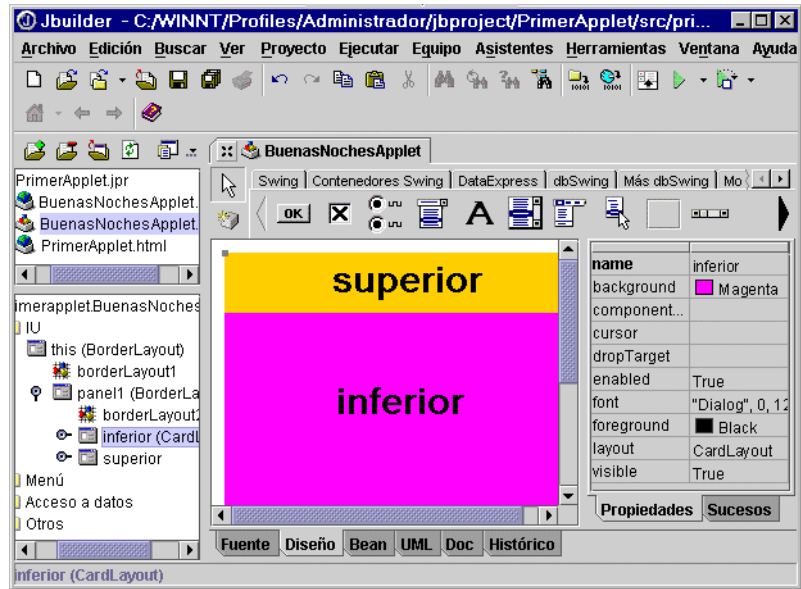
1 Seleccione el panel superior en el árbol de componentes o en el diseñador y haga doble clic en la columna que hay a la derecha de la propiedad `name` en el Inspector.

2 Escriba `superior` y pulse *Intro*.

Sugerencia

También se puede hacer clic en un componente con el botón derecho del ratón en el árbol de componentes y así elegir Cambiar nombre en el menú.

f Asigne al panel inferior el nombre inferior.



- 6 Cambie el color de fondo de superior a Orange en el Inspector:
 - a Seleccione superior en el árbol de componentes o el diseñador.
 - b Haga clic en la columna situada a la derecha de la propiedad background en el Inspector.
 - c Pulse la *flecha abajo* para abrir la lista desplegable de colores y seleccione Orange.

7 Cambie el color de fondo de inferior a Magenta.

8 Cambie el diseño de inferior a CardLayout.

El panel con CardLayout contendrá 5 paneles, cada uno de los cuales cada uno de los cuales mostrará con el texto “Buenas noches” en un idioma distinto. CardLayout coloca los componentes (normalmente paneles) uno encima de otro en una pila parecida a una baraja de cartas. Sólo se ven de uno en uno, y se pueden inspeccionar utilizando otro control para seleccionar qué panel se coloca en la parte superior. CardLayout se asocia normalmente a un componente de control, como una casilla de selección o una lista. El estado del componente de control determina qué componente presenta CardLayout. El usuario elige el estado mediante la selección de un elemento de la interfaz.

9 Añada cinco paneles (paneles 1 a 5) al panel inferior. Para ello, pulse *Mayús* mientras hace clic para seleccionar el componente Panel. Cada uno de estos paneles contendrá el texto “Buenas noches” en un idioma distinto. Elija la herramienta de selección para desactivar la selección múltiple de Panel.

Nota Si deja el componente en una ubicación errónea, selecciónelo en el árbol de componentes y pulse *Suprimir*. A continuación, añádalo de nuevo.

- 10 Cambie el diseño de los paneles del 1 al 5 a BorderLayout.
 - a Utilice *Ctrl+clic* o *Mayús+clic* para seleccionar los paneles del 1 al 5 en el árbol de componentes.
 - b Cambie la propiedad `layout` a BorderLayout en el Inspector. Los cinco paneles ya tienen el diseño BorderLayout.
 - c Seleccione otro componente del árbol de componentes o del diseñador de interfaces de usuario para que los paneles dejen de estar seleccionados.

- 11 Asigne a cada uno de los cinco paneles un color distinto en el Inspector.

Sugerencia Haga clic en la columna que se encuentra a la derecha de la propiedad `background` del Inspector, y pulse el botón de puntos suspensivos de la derecha de la lista desplegable para abrir el cuadro de diálogo Fondo. Si desea crear un color, utilice los controles de graduación del color del cuadro de diálogo Fondo.

- 12 Guarde el archivo y el proyecto.

- 13 Haga clic con el botón derecho del ratón en `BuenasNochesApplet.html` y seleccione Ejecutar utilizando “BuenasNochesApplet.”

Cuando se ejecuta el applet en el `appletviewer` de Sun sólo se ven el componente `superior` y el panel superior de `CardLayout`. Los paneles de los otros idiomas aparecen más adelante, después de añadir la lista desplegable y asociar los sucesos a las selecciones de la lista.

- 14 Salga del applet.

- 15 Cierre el panel de mensajes. Para ello, haga clic con el botón derecho del ratón en la pestaña `BuenasNochesApplet` y seleccione Eliminar pestaña “BuenasNochesApplet”.

Paso 5: Incorporación de componentes AWT al applet

Ahora se va a utilizar la paleta de componentes para añadir un componente `Label` y un componente `Choice` al panel superior del diseño, `superior`.



- 1 Pulse la pestaña AWT en la paleta de componentes y seleccione el componente `Choice`.
- 2 Coloque el componente en el panel naranja superior del diseño, `superior`. Utilice uno de los dos métodos siguientes:
 - Haga clic en el panel `superior` en el árbol de componentes.

- Haga clic en el panel superior en el diseñador de interfaces de usuario.

Observe que se ha añadido `choice1` a `superior` en el árbol de componentes.

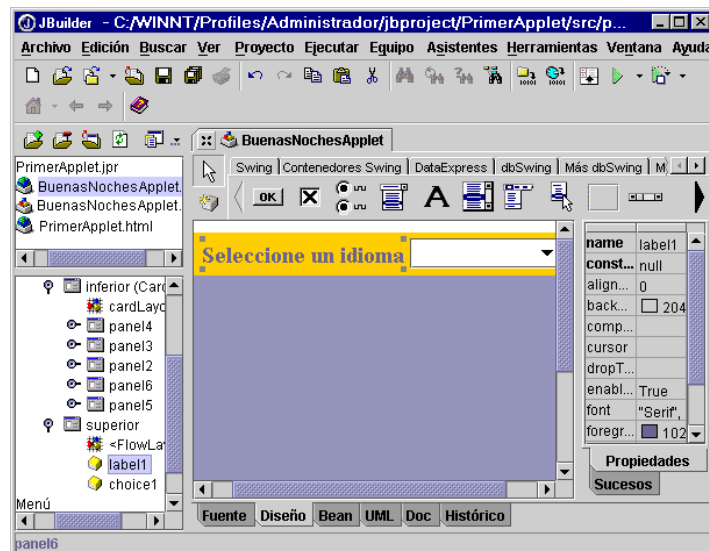


- 3 Seleccione el componente `Label` en la ficha AWT de la paleta de componentes y colóquelo en el panel superior, a la izquierda del componente `Choice`.

Observe que se ha añadido `label1` a `superior` en el árbol de componentes.

- 4 Seleccione `label1` en el árbol de componentes y haga lo siguiente:
 - a Haga doble clic en la columna de la derecha de la propiedad `text` del Inspector para resaltar el texto actual.
 - b Escriba `Seleccione un idioma` y pulse *Intro*. Ahora aparece “Seleccione un idioma” en la etiqueta adyacente al componente `Choice`.
 - c Para seleccionar el tipo de letra, haga clic en la columna situada a la derecha de la propiedad `font`. Pulse el botón de puntos suspensivos para abrir el cuadro de diálogo Fuente.
 - d Elija `Serif` en la lista de fuentes, seleccione `Negrita`. Escriba `20` en el cuadro Tamaño y pulse `Aceptar`.
 - e Para modificar el color del texto, haga clic en la columna situada a la derecha de la propiedad `foreground`. Pulse la *flecha abajo* y seleccione `Blue` en la lista desplegable de colores.

El diseño ofrecerá un aspecto similar a éste:



- 5 Pulse *Mayús* mientras hace clic para seleccionar el componente etiqueta AWT en la paleta de componentes, y añada una etiqueta a cada panel (del 1 al 5) en el panel inferior *CardLayout*.



En cada una de estas etiquetas pondrá “Buenas noches” en un idioma distinto. No olvide hacer clic en la herramienta de selección a la izquierda de la paleta de componentes, para desactivar la selección del componente.

- 6 Escriba “Buenas noches” en un idioma distinto en cada etiqueta. Primero, seleccione las etiquetas que hay debajo de los paneles en el árbol de componentes y escriba “Buenas noches” en el idioma correspondiente en la propiedad `text` del Inspector. Utilice estos idiomas para las etiquetas, o elija los que desee:
 - label2: Good Evening (inglés)
 - label3: Guten Abend (alemán)
 - label4: Oodgay vening eay (latín macarrónico)
 - label5: God Kväll (sueco)
 - label6: G'day, Mate (australiano)
- 7 Seleccione desde label2 hasta label6 con *Ctrl+clic*, y cambie la propiedad `font` de todas las etiquetas a **Bold** y el tamaño de fuente a 24. Cambie la propiedad `foreground` de las etiquetas a **Black**. Haga clic en un componente, en el árbol de componentes o en el diseñador de interfaces, para retirar la selección de las etiquetas.
- 8 Cambie la posición de las distintas etiquetas asignando en el Inspector los valores **North**, **South**, **East**, **West** y **Center** a sus propiedades `constraints`, como se indica a continuación.
 - label2: **North**
 - label3: **South**
 - label4: **East**
 - label5: **West**
 - label6: **Center**

Nota Observe la posición de cada etiqueta (como indican los tiradores de selección) en el panel *BorderLayout*. **Center** ocupa todo el panel, mientras que **North**, **South**, **East** y **West** sólo ocupan una parte.

- 9 Cambie el gestor de diseño de panel 5 a *FlowLayout*, el diseño utilizado para el panel superior.

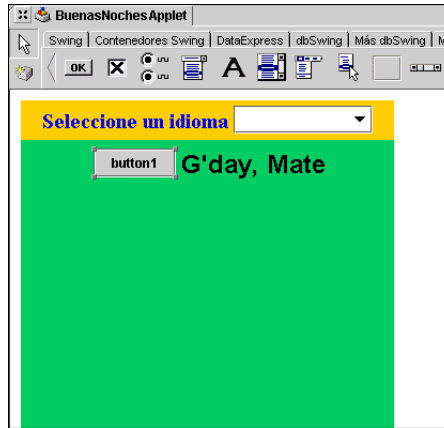
Observe que la etiqueta “G'day, Mate” se desplaza hasta el centro del borde superior del panel. *FlowLayout* es un diseño adecuado para disponer los componentes en hileras.

A continuación se añade un botón, y `FlowLayout` lo coloca en la misma fila que la etiqueta. `FlowLayout` coloca los componentes de izquierda a derecha en una fila. Cuando la fila está llena, los componentes que no caben en ella pasan a la siguiente.



- 10 Haga clic en el componente `Button` de AWT, arrástrelo y suéltelo a la izquierda de la etiqueta "G'day, Mate".

Pruebe a desplazar el botón a otra posición en el diseñador. `FlowLayout` siempre lo devuelve a la fila en la que se encuentra la etiqueta. El aspecto del applet para `panel5` debe ser parecido a éste:



- 11 Seleccione Archivo | Guardar todo para guardar el proyecto.

Paso 6: Modificación del código fuente

En este paso se añaden los lenguajes a la lista desplegable y después se añaden sucesos asociados a los paneles de lenguajes en el componente `Choice`.

- 1 Añada los idiomas de la lista desplegable al método `init()` de esta forma:
 - a Seleccione la pestaña Fuente en el panel de contenido para dirigirse al código fuente en el editor.
 - b Seleccione el método `init()` en el panel de estructura. El código del método `init()` está resaltado en el editor.

Sugerencia

Busque un método en el panel de estructura haciendo clic en el panel y escribiendo su nombre.

- c Sitúe el cursor entre la primera llave y la sentencia `try/catch` y pulse *Intro* para añadir una línea en blanco.

Sugerencia

Para ampliar el editor y ocultar el proyecto y el panel de estructura, seleccione **Vista | Conmutar**.

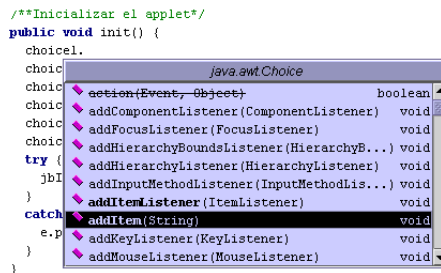
- d** Añada el siguiente bloque de código, marcado en negrita, al método `init()`:

```
//Inicializar el applet
public void init() {
    choice1.addItem("iEspañolInglés");
    choice1.addItem("alemán");
    choice1.addItem("latín macarrónico");
    choice1.addItem("sueco");
    choice1.addItem("inglés australiano");

    try {
        jbInit();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}
```

Sugerencia

Utilice **CodeInsight** para completar el código. Escriba `choice1.` y espere a que aparezca la ventana emergente, o pulse **Ctrl+Espacio** para abrirla. No olvide incluir el punto (.) después de `choice`. Utilice las flechas para seleccionar `addItem(String)` en la ventana emergente. Pulse **Intro**. Puede configurar **CodeInsight** en el cuadro de diálogo **Opciones del editor (Herramientas | Opciones del editor | CodeInsight)**.



Si el código contiene algún error de sintaxis, aparece la carpeta **Errores** del panel de estructura mientras se escribe en el editor. Abra la carpeta y seleccione el mensaje de error para resaltar el error en el código fuente.

Consulte

- “Acerca de los mensajes de advertencia y error” en *Creación de aplicaciones con JBuilder*.

A continuación conectará el suceso a la elección del idioma. Cuando se selecciona un idioma del componente lista desplegable `Choice` de la lista desplegable, en el panel `cardLayout` aparece “Buenas noches” en el idioma seleccionado.

2 Asocie los sucesos de la lista `Choice` de la siguiente manera:

- a Vuelva al diseñador de interfaces de usuario.
- b Seleccione `choice1`, situado bajo `superior` en el árbol de componentes.
- c Seleccione la pestaña Sucesos en el Inspector.
- d Haga doble clic a la derecha del suceso `itemStateChanged`.

`JBuilder` genera el código del método y lleva el cursor al método correspondiente en el código fuente.

```
void choice1_itemStateChanged(ItemEvent e) {
}
```

- e Añada el siguiente código, marcado en negrita, para asociar los paneles de idioma a los sucesos de elección del idioma:

```
void choice1_itemStateChanged(ItemEvent e) {
    if ("Español".equals(choice1.getSelectedItem())){
        cardLayout1.show(lower, "panel1");
    }
    else if ("alemán".equals(choice1.getSelectedItem())){
        cardLayout1.show(lower, "panel2");
    }
    else if ("latín macarrónico".equals(choice1.getSelectedItem())){
        cardLayout1.show(lower, "panel3");
    }
    else if ("sueco".equals(choice1.getSelectedItem())){
        cardLayout1.show(lower, "panel4");
    }
    else if ("inglés australiano".equals(choice1.getSelectedItem())){
        cardLayout1.show(lower, "panel5");
    }
}
```

Sugerencia

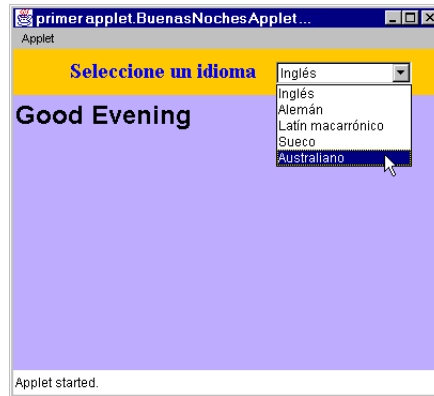
Puede utilizar plantillas para generar el código. Escriba `if` y pulse ***Ctrl+J*** para acceder a la ventana emergente de plantillas de código. Utilice las teclas de flecha para desplazarse por las selecciones. Seleccione la plantilla `if-else if` y pulse ***Intro***. Se genera el código:

```
if () {
}
else if{
}
}
```

3 Elija Archivo | Guardar todo.

- 4 Ejecute el applet. Para ello, haga clic con el botón derecho en `BuenasNochesApplet.html` en el panel del proyecto y seleccione Ejecutar utilizando “BuenasNochesApplet”.

El applet “Buenas Noches” se ejecuta en el `appletviewer` de Sun:



Si surge algún error, aparece en el panel de mensajes, en la parte inferior del Visualizador de aplicaciones. Si desea ayuda sobre un mensaje de error, selecciónelo y pulse *F1*. Seleccione el mensaje de error para resaltar el código correspondiente en el editor. En ocasiones, el error puede encontrarse antes o después de la línea de código resaltada. Corrija los errores, guarde el proyecto y ejecute el applet de nuevo.

- 5 Pruebe la lista desplegable. El idioma seleccionado en la lista debe coincidir con el del panel situado debajo.
- 6 Salga del applet.

Ahora se van a añadir una etiqueta y un suceso de botón para `button1` en el panel 5. Al pulsar el botón, el texto “G’day, Mate” de la etiqueta `label6` se vuelve rojo.

- 1 Añada una etiqueta y un suceso de botón de la forma que se explica a continuación:
 - a Vuelva al diseñador de interfaces de usuario.
 - b Seleccione `button1` en panel 5 y cambie la propiedad `Label` del botón en la ficha `Propiedades del Inspector`, de `button1` a `Púlsame`. Pulse *Intro*. Observe que el botón cambia de tamaño automáticamente para adaptarse al texto.
 - c Haga clic en la pestaña `Sucesos del Inspector` para definir lo que debe ocurrir al pulsar `button1`.
 - d Haga doble clic en la columna que se encuentra a la derecha del suceso `ActionPerformed`.

JBUILDER pasa al editor en el que se ha añadido el siguiente código de estructura para el suceso `ActionPerformed`, a continuación de las sentencias `if-else if`.

```
void button1_actionPerformed(ActionEvent e) {  
  
}
```

Sugerencia

El doble clic en el botón en el diseñador tiene el mismo efecto.

Ahora escriba el código que define el suceso del botón que cambia a rojo el texto “G’day, Mate”.

- e Escriba el siguiente código, marcado en negrita:

```
void button1_actionPerformed(ActionEvent e) {  
    label1.setForeground(new Color(255,0,0));  
}
```

- 2 Guarde el proyecto.
- 3 Ejecute el applet y seleccione “Australiano” en la lista desplegable. Pulse el botón “Púlsame”.

“G’day, Mate” aparece en rojo.

El applet presentará un aspecto similar a éste:



- 4 Salga del applet.

Paso 7: Distribución del applet

La distribución de un applet Java consiste en reunir los diferentes archivos de clases Java, imágenes y demás archivos necesarios para el applet y copiarlos, junto con el archivo HTML del applet, en una dirección de un ordenador cliente o servidor para que se puedan ejecutar. Los archivos se pueden entregar por separado, o todos juntos en un recopilatorio comprimido o sin comprimir. Los recopilatorios más utilizados son los

archivos de Java, JAR. Los archivos JAR cuentan con la ventaja de que su tamaño es más pequeño, por lo que el tiempo de descarga se reduce.

Cuando distribuya el applet, es importante que recuerde lo siguiente:

- Guarde y compile el proyecto antes de la distribución.
- Abra `BuenasNochesApplet.html` y pulse la pestaña Fuente para revisar lo siguiente:
 - Compruebe que el atributo `code` tiene el nombre completo de la clase, incluido el nombre de paquete: En este ejemplo, el valor de `code` es `primerapplet.BuenasNochesApplet.class`.
 - Compruebe que el atributo `codebase` de `BuenasNochesApplet.html` especifica la ubicación correcta del archivo de clase con relación al archivo HTML. En este ejemplo, el valor de `codebase` es `"."`, porque el recopilatorio JAR que contiene el archivo de clase estará en el mismo directorio que el archivo HTML.
- Conserve la estructura actual de directorios. En este ejemplo, `BuenasNochesApplet.class` debe estar en el directorio `primerapplet` para reflejar la estructura de paquete: `primerapplet/BuenasNochesApplet.class`. Si va a realizar la distribución con un archivo JAR, compruebe su estructura de directorios y cerciórese de que coincide.
- Envíe todas las clases necesarias al servidor de Internet adecuado. Los archivos de clase deben estar en la ubicación correcta respecto al archivo HTML y deben coincidir con el atributo `codebase`.
- Envíe el archivo HTML del applet al servidor de Internet adecuado.

Advertencia Si está creando los applets para navegadores compatibles con la versión JDK 1.02, recuerde que estos navegadores no admiten los archivos JAR. En su lugar, cree un archivo ZIP.

Consulte

- “Depuración de programas en Java”, en *Creación de aplicaciones con JBuilder*.
- Paso 16 del tutorial de JBuilder “Distribución de la aplicación Editor de texto” en *Diseño de aplicaciones con JBuilder*.

Según la edición de JBuilder, existen varias herramientas para distribuir el applet:

- La herramienta `jar` de Java, disponible con el JDK.
- El Creador de recopilatorios de JBuilder, disponible en SE y Enterprise.

Distribución de applets con la herramienta Jar

JBuilder Personal

El JDK incluye en el directorio `bin` una herramienta `jar` que permite crear archivos JAR para la distribución. La herramienta `jar` es una herramienta de recopilación y compresión que reúne varios archivos en un archivo recopilatorio JAR.

El comando `jar` básico tiene la siguiente forma:

```
jar {ctxu}[vfmOM] [archivo jar] [archivo descriptor] [-C dir] archivos ...
```

Nota Si desea ayuda sobre más opciones de JAR, escriba `jar -help` en la línea de comandos o consulte la documentación sobre `jar` en <http://java.sun.com/j2se/1.3/docs/tool docs/tools.html#basic>.

Cree el archivo JAR mediante estos pasos:

- 1 Guarde el proyecto.
- 2 Seleccione Proyecto | Ejecutar Make del proyecto "PrimerApplet.jpx" para compilar el proyecto.
- 3 Cree un subdirectorio `applets` para el applet en el directorio `<inicio>/jbproject/`.

Éste será el directorio de prueba donde pondrá el archivo HTML del applet y el archivo JAR.

- 4 Abra el shell de línea de comandos o la ventana de DOS.
- 5 Diríjase al directorio del proyecto:
`<inicio>/jbproject/PrimerApplet/`
- 6 Escriba el siguiente comando JAR.

```
jar cvf BuenasNoches.jar -C classes primerapplet
```

Donde

<code>jar</code>	es el comando que genera el archivo JAR
<code>.c</code>	crea un archivo recopilatorio
<code>v</code>	genera la salida e indica lo que se ha añadido al archivo JAR
<code>f</code>	especifica el nombre del archivo JAR que se va a crear
<code>C</code>	cambia provisionalmente al directorio especificado durante la ejecución del comando <code>jar</code> (en este caso, el directorio <code>classes</code>)
<code>classes</code>	es el directorio en el que se ejecuta el comando <code>jar</code>
<code>primerapplet</code>	es el paquete de las clases que se van a añadir al archivo JAR

Nota El JDK debe encontrarse en su vía de acceso. Si no es así, utilice el siguiente comando:

```
/<jbuilder>/<jdk>/bin/jar cvf BuenasNoches.jar -C classes primerapplet
```

Donde <jbuilder> es el nombre del directorio de JBuilder y <jdk> es el nombre del directorio del JDK. Por ejemplo: jbuilder6/jdk1.3.1/... Si JBuilder está en otra unidad, incluya su nombre antes del directorio de JBuilder.

Nota Para Windows, utilice barras invertidas (\).

La herramienta jar muestra en la pantalla el resultado, que debe tener el siguiente aspecto:

```
added manifest
adding: primerapplet/(in = 0) (out= 0) (stored 0%)
adding: primerapplet/BuenasNochesApplet.class(in = 5501) (out= 2683)
      (deflated 51%)
adding: primerapplet/BuenasNochesApplet$1.class(in = 745) (out= 411)
      (deflated 44%)
adding: primerapplet/BuenasNochesApplet$2.class(in = 741) (out= 409)
      (deflated 44%)
```

Observe la estructura de directorios de las clases:

primerapplet/BuenasNochesApplet.class. Además, observe que la herramienta jar ha creado un archivo descriptor. La metainformación sobre el recopilatorio se almacena en el archivo descriptor. El archivo JAR se guarda en el directorio del proyecto PrimerApplet.

7 Copie BuenasNoches.jar al directorio applets para someterlo a prueba.

Después de crear el archivo JAR, continúe con el [“Paso 8: Modificación del archivo HTML” en la página 9-32](#).

Consulte

- “Using JAR Files: The Basics” en <http://java.sun.com/docs/books/tutorial/jar/basics/index.html>.
- “jar-The Java Archive Tool” en <http://java.sun.com/j2se/1.3/docs/tool docs/tools.html#basic>.
- “Depuración de programas en Java”, en *Creación de aplicaciones con JBuilder*.

Distribución del applet con el Creador de recopilatorios

Es una función de
JBuilder SE y Enterprise.

El Creador de recopilatorios de JBuilder reúne todos los archivos necesarios para distribuir el applet y los guarda en un archivo JAR.

Para distribuir el applet con JBuilder SE y Enterprise:

1 Guarde el proyecto.

- 2 Seleccione Proyecto | Ejecutar Make del proyecto “PrimerApplet.jpj” para compilar el proyecto.
- 3 Cree un subdirectorio applets para el applet en el directorio /<inicio>/jbproject/.
Éste será el directorio de prueba donde pondrá el archivo HTML del applet y el archivo JAR.
- 4 Elija Asistentes | Creador de recopilatorios para abrir el Creador de recopilatorios.



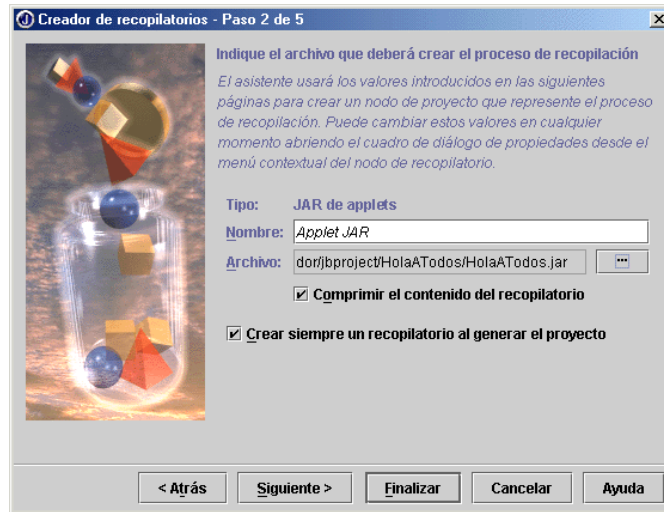
- 5 Seleccione Applet JAR en la lista desplegable Tipos de recopilatorio del Paso 1. Pulse Siguiente para avanzar al Paso 2.

Nota

Seleccione el tipo de archivo ZIP si está desarrollando applets para navegadores antiguos que no acepten los archivos JAR.

- 6 Acepte el nombre por defecto de Applet JAR en el campo Nombre.
- 7 Pulse el botón de puntos suspensivos contiguo al campo Archivo y desplácese al directorio /<inicio>/jbproject/applets/.
- 8 Cambie el nombre del archivo JAR en el campo Nombre del archivo por el de BuenasNoches.jar y pulse Aceptar.

El Paso 2 debería tener este aspecto.



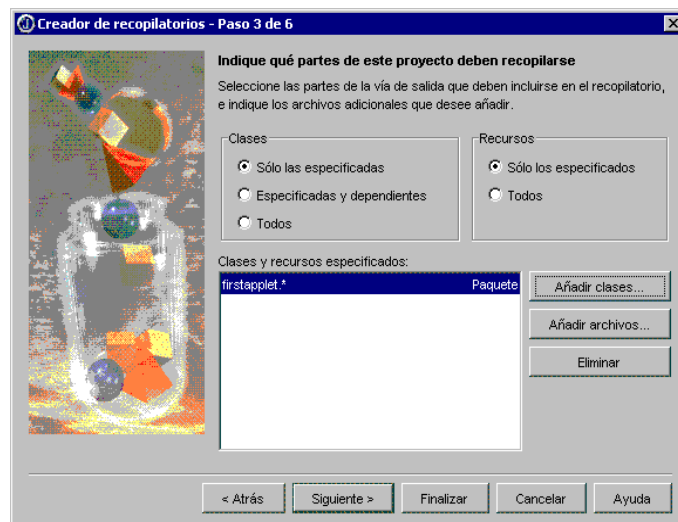
9 Pulse Siguiente para pasar al Paso 3.

10 En las opciones de Clases, seleccione Sólo las especificadas.

Elija de forma manual las clases que desea añadir. Esta opción permite excluir el archivo HTML, ya que no es necesario que se distribuya con las clases.

11 Pulse Añadir clases y seleccione el paquete `primerapplet`. Haga clic en Aceptar para cerrar el cuadro de diálogo.

El paquete se muestra en la lista Clases y recursos especificados.



12 Pulse Siguiente para pasar al Paso 4.

13 Pulse Siguiente para pasar al Paso 5.

No es necesario añadir ninguna biblioteca.

14 Acepte las opciones por defecto del Paso 5.

Observe que está activada la opción de crear un archivo descriptor para el recopilatorio.

Nota Si desea obtener más información sobre los archivos descriptors, consulte “Sobre los archivos descriptors” en *Creación de aplicaciones con JBuilder*.

15 Pulse Finalizar para salir del Creador de recopilatorios.

En el panel del proyecto aparece un nodo de archivo denominado Applet JAR. Para modificar este archivo haga clic en él con el botón derecho y seleccione Propiedades.

16 Haga clic con el botón derecho del ratón en el nodo del recopilatorio Applet JAR y seleccione Ejecutar Make para compilar el proyecto.

El Creador de recopilatorios recoge el paquete `primerapplet`, que se ha añadido al archivo JAR en el tercer paso del asistente.

17 Pulse el icono de ampliación situado junto al nodo del archivo Applet JAR para ver el recopilatorio `BuenasNoches.jar`.

18 Haga doble clic en el archivo JAR, en el panel del proyecto.

El archivo descriptor aparece en el panel de contenido, y el contenido del archivo JAR se muestra en el panel de estructura. Seleccione un archivo en el panel de estructura para verlo en el panel de contenido.

Nota Si distribuye varios programas en la misma ubicación, puede entregar los archivos redistribuibles de forma separada, antes que incluirlos en cada uno de los archivos JAR.

Consulte

- “Distribución con el Creador de recopilatorios” en *Creación de aplicaciones con JBuilder*.
- “Depuración de programas en Java”, en *Creación de aplicaciones con JBuilder*.

Paso 8: Modificación del archivo HTML

Ahora que se ha distribuido el applet en un archivo JAR es necesario modificar el archivo HTML con el atributo `archive` e incluir el nombre del archivo JAR. También se añadirá, dentro de las etiquetas `<applet>`, un mensaje que comunique a los usuarios que no tengan navegadores compatibles con Java que no podrán ver el applet a menos que activen Java en el navegador o actualicen la versión.

Para modificar un archivo HTML,

- 1 Abra `BuenasNochesApplet.html` en JBuilder y añada el atributo `archive`:

a Seleccione la pestaña Fuente para ver el código fuente HTML.

- b** Añada el siguiente código HTML dentro de la etiqueta `<applet>`:

```
archive = "BuenasNoches.jar"
```

La etiqueta `<applet>` tiene este aspecto:

```
<applet
  codebase = "."
  code     = "primerapplet.GoodEveningApplet.class"
  archive  = "GoodEvening.jar"
  name     = "TestApplet"
  width    = 400
  height   = 300
  hspace   = 0
  vspace   = 0
  align    = top
>
</applet>
```

Sugerencia

Si tiene varios archivos JAR para el applet, haga una lista de ellos separándolos con comas, como se indica a continuación:

```
archive="archivo1.jar, archivo2.jar"
```

Importante

Algunos de los navegadores web más antiguos no son compatibles con los archivos JAR ni con los listados de múltiples recopilatorios, aunque sí admiten un archivo ZIP único en el atributo `archive`.

A continuación se va a añadir un mensaje que comunica a los usuarios que no tengan Java activado que su navegador no es compatible con Java, por lo que no pueden ver el applet.

- 2 Escriba el siguiente mensaje entre las etiquetas `<applet>` de apertura y cierre:

```
Para ver este applet se necesita un navegador compatible con Java que
utilice el JDK 1.1.x o posterior.
```

La etiqueta `<applet>` tiene este aspecto:

```
<applet
  codebase = ". "
  code     = "primerapplet.GoodEveningApplet.class"
  archive  = "GoodEvening.jar"
  name     = "TestApplet"
  width    = 400
  height   = 300
  hspace   = 0
  vspace   = 0
  align    = top
>
```

Necesitará un navegador Java que utilice JDK 1.1.x o superior para ver este applet.

```
</applet>
```

Los navegadores que no son compatibles con Java obvian las etiquetas `<applet>` y muestran todo lo que se encuentra entre ellas. Dado que los navegadores compatibles con Java reconocen las etiquetas `<applet>`, los usuarios que tengan Java activado verán el applet y no el mensaje.

Importante

Antes de guardar el archivo HTML, compruebe de nuevo los valores de `codebase` y de `code`. Si estos valores son incorrectos, el applet no se ejecutará. Recuerde que el valor `codebase` indica la ubicación del código del applet (archivo CLASS o JAR) respecto al archivo HTML. El valor, `". "`, significa que el archivo de clase se encuentra en el mismo directorio que el archivo HTML. El valor de `code` debe ser el nombre de clase completo del applet, incluido el nombre de paquete.

3 Guarde y cierre el archivo.

4 Copie el archivo `AppletBuenasNoches.html` modificado del directorio `classes` del proyecto al directorio `applets`. El directorio `applets` debe incluir dos archivos, `BuenasNochesApplet.html` y `BuenasNoches.jar`.

Advertencia

Recuerde que JBuilder crea dos archivos HTML: el de notas de proyecto, `PrimerApplet.html`, que se encuentra en el directorio raíz del proyecto, y el del applet, `BuenasNochesApplet.html`, que contiene la etiqueta `<applet>` y se encuentra en el directorio `src` del proyecto. Si se copia `PrimerApplet.html` en vez de `BuenasNochesApplet.html` en el directorio `applets`, el applet no se ejecuta.

Paso 9: Ejecución del applet distribuido desde la línea de comandos

Es aconsejable probar localmente el applet distribuido antes de probarlo en la Web. Puede hacerlo desde la línea de comandos, por medio del `appletviewer` de Sun. De esta forma puede comprobar si el navegador tiene todo lo necesario para ejecutar el applet. Si falta algún archivo o si existe algún error en el archivo HTML, el applet no se ejecutará. De este modo puede corregir los errores antes de enviar el applet a la Web.

Para ejecutar el applet desde la línea de comandos:

- 1 Cerciórese de que hay copias de los archivos `BuenasNochesApplet.html` y `BuenasNoches.jar` en el directorio `applets`.
- 2 Abra la ventana de línea de comandos.
- 3 Elimine todas las variables `CLASSPATH` para borrar todos los valores de vías de acceso a clases de esta sesión, de la siguiente manera:
 - Windows NT, 2000 y XP: `set CLASSPATH=`
 - Unix:
 - en el shell `csh`: `unsetenv CLASSPATH`
 - en el shell `sh`: `unset CLASSPATH`
- 4 Sitúese en el directorio `applets/`.
- 5 Ejecute el `appletviewer` mediante el siguiente comando:

```
<jbuilder>/<jdk>/bin/appletviewer BuenasNochesApplet.html
```

Donde `<jbuilder>` representa la versión de JBuilder que se ejecuta, por ejemplo, `jbuilder6/`.

Importante Si JBuilder está en otra unidad, incluya la letra de la unidad.

Nota Para Windows, utilice barras invertidas (`\`).

Si el applet “Buenas noches” se carga y se ejecuta en el `appletviewer`, la distribución es correcta, y todas las clases se han encontrado y se han incluido. Si el applet no se ejecuta, compruebe los mensajes de error, corríjalos, vuelva a compilar, distribuya y haga otra prueba.

En el subdirectorio `samples/tutorials/FirstPrimerApplet/` del directorio de la instalación de JBuilder hay un ejemplo del applet “Good Evening”, en inglés.

Si experimenta problemas con la ejecución, compruebe el código fuente del applet al final del tutorial y consulte estos temas para resolver errores frecuentes:

- “Solving common applet problems” en <http://www.java.sun.com/docs/books/tutorial/applet/problems/index.html>

- “Errores habituales en la etiqueta <applet>” y “Consejos adicionales para que funcionen las applets” en la *Guía del desarrollador de aplicaciones web*.

El último paso del desarrollo del applet consiste en probarlo ejecutándolo en Internet. Así sabrá si de verdad contiene todos los archivos que necesita. Aunque este tutorial no entra en detalles sobre la comprobación de applets, consulte [“Comprobación del applet distribuida en la web” en la página 9-35](#) para obtener una descripción general de la comprobación de applets.

¡Enhorabuena! Ha creado su primer applet con JBuilder. Ahora que ya se ha familiarizado con el entorno de desarrollo de JBuilder, descubrirá que sus múltiples funciones le permiten ahorrar tiempo y programar con mayor comodidad.

Si desea ver otros tutoriales relacionados con los applets, consulte:

- “The Java Tutorial” en <http://java.sun.com/docs/books/tutorial/index.html>
- Charlie Calvert, Part II: Applets en <http://homepages.borland.com/ccalvert/JavaCourse/index.htm>
- La página web Curmudgeon de Rich Wilkman, en <http://www.formlessvoid.com/jc/index>
- Página web de John Moore, “Applet design and deployment”, en http://www.microps.com/mps/p_appletdesign.html

Comprobación del applet distribuida en la web

El último paso de la distribución consiste en probar el applet ejecutándolo en la Web. Así sabrá si de verdad contiene todos los archivos que necesita. Es importante comprobar el applet en varios navegadores distintos para asegurarse de que el compilador contiene los archivos necesarios. Aunque este tutorial no profundiza en la comprobación de applets, a continuación se detallan algunas instrucciones de comprobación.

Lleve a cabo estos pasos y, a continuación, compruebe el applet ejecutándolo en varios navegadores web:

- 1 Transfiera los archivos HTML y JAR del applet a un servidor de Internet o cópielos a un servidor de Windows NT.
 - a Emplee una utilidad de FTP para transferir los archivos al servidor. No olvide indicar que se trata de archivos binarios.
 - b Compruebe que la posición de los archivos HTML y JAR en el servidor coincide con el atributo `codebase` del archivo HTML y que el atributo `code` tiene el nombre de clase completo, incluido el nombre del paquete.

- 2 Pruebe el applet con diferentes navegadores. Si el applet no se carga, compruebe que el navegador es compatible con Java. También compruebe la Consola Java del navegador por si hubiera mensajes de error.

Para abrir la Consola Java:

- Seleccione Communicator | Herramientas | Consola Java en Netscape.
 - Seleccione Ver | Consola Java en Internet Explorer.
- 3 Corrija todos los errores, vuelva a distribuir el applet y compruébelo de nuevo en los navegadores.

Consulte

- “Comprobación de las applets” en la *Guía del desarrollador de aplicaciones web*.

Código fuente del applet

Código fuente HTML del applet BuenasNochesApplet.html

```
<html>
<head>
<meta http="Content-Type" content="text/html; charset=windows-1252">
<title>
Página HTML de BuenasNoches
</title>
</head>
<body>
primerapplet.GoodEveningApplet will appear below in a Java enabled browser.
```

```
<applet
  codebase = "."
  code     = "primerapplet.GoodEveningApplet.class"
  archive  = "GoodEvening.jar"
  name     = "TestApplet"
  width    = 400
  height   = 300
  hspace   = 0
  vspace   = 0
  align    = top
>
Necesitará un navegador Java que utilice JDK 1.1.x o superior para ver este applet.
</applet>
</body>
</html>
```

Código fuente de la clase del applet BuenasNochesApplet.java

```

package primerapplet;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;

public class GoodEveningApplet extends Applet {
    boolean isStandalone = false;
    BorderLayout BorderLayout1 = new BorderLayout();
    Panel lower = new Panel();
    Panel upper = new Panel();
    CardLayout cardLayout1 = new CardLayout();
    Panel panel1 = new Panel();
    Panel panel2 = new Panel();
    Panel panel3 = new Panel();
    Panel panel4 = new Panel();
    Panel panel5 = new Panel();
    BorderLayout BorderLayout2 = new BorderLayout();
    BorderLayout BorderLayout3 = new BorderLayout();
    BorderLayout BorderLayout4 = new BorderLayout();
    BorderLayout BorderLayout5 = new BorderLayout();
    Choice choice1 = new Choice();
    Label label1 = new Label();
    Label label2 = new Label();
    Label label3 = new Label();
    Label label4 = new Label();
    Label label5 = new Label();
    Label label6 = new Label();
    FlowLayout flowLayout1 = new FlowLayout();
    Button button1 = new Button();
    /**Obtener un valor de parámetro*/
    public String getParameter(String key, String def) {
        return isStandalone ? System.getProperty(key, def) :
            (getParameter(key) != null ? getParameter(key) : def);
    }

    /**Construir el applet*/
    public GoodEveningApplet() {
    }
    /**Inicializar el applet*/
    public void init() {
        choice1.addItem("i Español nglés");
        choice1.addItem("alemán");
        choice1.addItem("latín macarrónico");
        choice1.addItem("sueco");
        choice1.addItem("inglés australiano");
    }
}

```

```

        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
}
/**Iniciación de componente*/
private void jbInit() throws Exception{
    this.setLayout (borderLayout1);
    upper.setBackground(Color.orange);
    lower.setBackground(Color.magenta);
    lower.setLayout(cardLayout1);
    panel1.setLayout(borderLayout2);
    panel2.setLayout(borderLayout3);
    panel3.setLayout(borderLayout4);
    panel4.setLayout(borderLayout5);
    panel5.setLayout(flowLayout1);
    panel1.setBackground(new Color(190, 173, 255));
    panel2.setBackground(new Color(83, 182, 255));
    panel3.setBackground(new Color(255, 149, 66));
    panel4.setBackground(new Color(239, 107, 140));
    panel5.setBackground(new Color(17, 198, 99));
    label1.setFont(new java.awt.Font("Serif", 1, 20));
    label1.setForeground(Color.blue);
    label1.setText("Seleccione un idioma");
    label2.setFont(new java.awt.Font("Dialog", 1, 24));
    label2.setForeground(Color.black);
    label2.setText("Good Evening");
    label3.setFont(new java.awt.Font("Dialog", 1, 24));
    label3.setForeground(Color.black);
    label3.setText("Guten Abend");
    label4.setFont(new java.awt.Font("Dialog", 1, 24));
    label4.setForeground(Color.black);
    label4.setText("Oodgay vening eay");
    label5.setFont(new java.awt.Font("Dialog", 1, 24));
    label5.setForeground(Color.black);
    label5.setText("God Kvall");
    choice1.addItemListener(new java.awt.event.ItemListener() {
        public void itemStateChanged(ItemEvent e) {
            choice1_itemStateChanged(e);
        }
    });
    label6.setFont(new java.awt.Font("Dialog", 1, 24));
    label6.setForeground(Color.black);
    label6.setText("G' day, Mate");
    button1.setLabel("Pulse me");
    button1.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(ActionEvent e) {
            button1_actionPerformed(e);
        }
    });
    this.add(lower, BorderLayout.CENTER);
}

```

```

lower.add(panel1, "panel1");

panel1.add(label2, BorderLayout.NORTH);
lower.add(panel2, "panel2");
panel2.add(label3, BorderLayout.SOUTH);
lower.add(panel3, "panel3");
panel3.add(label4, BorderLayout.EAST);
lower.add(panel4, "panel4");
panel4.add(label5, BorderLayout.WEST);
lower.add(panel5, "panel5");
panel5.add(button1, null);
panel5.add(label6, null);
this.add(upper, BorderLayout.NORTH);
upper.add(label1, null);
upper.add(choice1, null);
}
/**Iniciar el applet*/
public void start() {
}
/**Detener el applet*/
public void stop() {
}
/**Destruir el applet*/
public void destroy() {
}
/**Obtener información del applet*/
public String getAppletInfo() {
    return "Applet Information";
}
/**Obtener información del parámetro*/
public String[][] getParameterInfo() {
    return null;
}

void choice1_itemStateChanged(ItemEvent e) {
    if ("Español".equals(choice1.getSelectedItem())){
        cardLayout1.show(lower, "panel1");
    }
    else if ("alemán".equals(choice1.getSelectedItem())){
        cardLayout1.show(lower, "panel2");
    }
    else if ("latín macarrónico".equals(choice1.getSelectedItem())){
        cardLayout1.show(lower, "panel3");
    }
    else if ("sueco".equals(choice1.getSelectedItem())){
        cardLayout1.show(lower, "panel4");
    }
    else if ("inglés australiano".equals(choice1.getSelectedItem())){
        cardLayout1.show(lower, "panel5");
    }
}

void button1_actionPerformed(ActionEvent e) {
    label6.setForeground(new Color(255, 0, 0));
}
}

```


Tutorial: La vista Histórico

En este tutorial se explica el uso de la vista Histórico. La vista Histórico está disponible en todas las ediciones de JBuilder, aunque las funciones pueden variar de una a otra. Este tutorial hace referencia a todas las funciones disponibles, organizadas por ediciones. Los usuarios de todas las ediciones de JBuilder pueden completar las dos primeras lecciones. Los usuarios de JBuilder SE y Enterprise pueden completar las tres primeras. Los usuarios de JBuilder Enterprise pueden completar las cuatro lecciones. Los cambios realizados en este tutorial son extremadamente sencillos, con el fin de que pueda centrarse más en las funciones del Histórico que en el código.

La terminología que describe la gestión de las versiones varía según la herramienta. En este tutorial se utilizan términos genéricos para las utilidades de gestión de versiones de JBuilder independientes de la herramienta. Para ver las explicaciones de estos términos, consulte [“Glosario de gestión de versiones” en la página 8-1](#).

El uso de los sistemas de control de versiones se analiza en los casos necesarios. El tutorial se puede utilizar con o sin sistema de control de versiones (los pasos que se aplican sólo a los usuarios de sistemas de control de versiones se indican claramente). La integración con el sistema de control de versiones Concurrent Versions System (CVS) es una función de JBuilder SE y Enterprise. Las integraciones con Microsoft Visual SourceSafe (VSS) y Rational ClearCase son funciones de JBuilder Enterprise. Si desea obtener más información sobre el control de versiones en JBuilder, consulte *Desarrollo en equipo con JBuilder*. Si desea obtener información específica sobre CVS, consulte “CVS en JBuilder” en *Desarrollo en equipo con JBuilder*.

Consulte el apartado Tutoriales en las Sugerencias de JBuilder si desea información sobre cómo ver e imprimir tutoriales. El apartado Opciones de accesibilidad de Sugerencias de JBuilder contiene sugerencias sobre la utilización de las funciones de JBuilder para mejorar la facilidad de uso de JBuilder para personas con discapacidades.

Para obtener información sobre las convenciones utilizadas en este tutorial y en otra documentación de JBuilder, consulte [“Convenciones de la documentación” en la página 2-4.](#)

Paso 1: Generación de varias versiones de un archivo

En primer lugar es necesario generar varias versiones diferentes de un archivo, de manera que haya algo que ver en la vista Histórico. A lo largo de este tutorial, trabajará con archivos del proyecto de ejemplo Bienvenida.

Para realizar cambios en un archivo fuente:

- 1 Abra el proyecto de ejemplo Bienvenida.

El ejemplo, `Wel come.j px`, está situado en el subdirectorio `samples/wel come` del directorio de instalación de JBuilder.

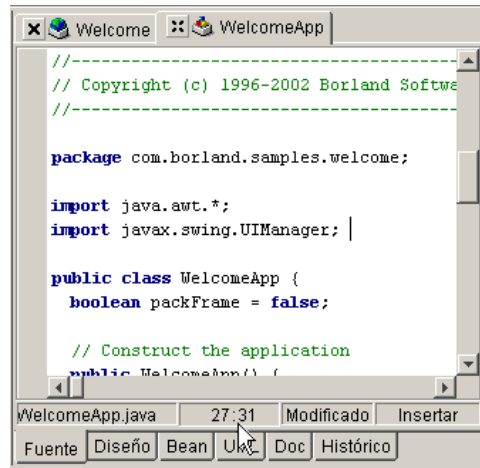
Nota El siguiente paso sólo es aplicable a usuarios de sistemas de control de versiones.

- 2 Si utiliza un sistema de control de versiones: coloque el proyecto bajo el control de versiones.

Nota Si no tiene los derechos de acceso necesarios para colocar un proyecto en el sistema de control de versiones pero desea conocer las funciones de control de versiones que ofrece la vista Histórico, puede utilizar la instalación de CVS incluida en JBuilder. CVS permite crear un repositorio local controlado completamente por el usuario. Si desea obtener más información sobre el control de versiones en JBuilder, consulte *Desarrollo en equipo con JBuilder*. Si desea obtener información específica sobre CVS, consulte “CVS en JBuilder” en *Desarrollo en equipo con JBuilder*.

- 3 Haga doble clic en `AplicaciónBienvenida.java`, en el panel de proyecto, si aún no está abierto y activo en el panel de contenido.
- 4 Observe la barra de estado del panel de contenido.

Los dos números de la barra de estado indican la línea y la columna de la posición actual del cursor.



5 Ponga el cursor al final de la línea 27. El texto de esta línea es `import javax.swing.UIManager;`.

6 Pulse *Intro* al final de la línea 27 para añadir una línea por debajo.

7 Escriba `import java.applet.*;` en la fila 28.



8 Seleccione Archivo | Guardar archivo o pulse el icono Guardar archivo para guardar el archivo.

Esto genera una versión de copia de seguridad que se verá posteriormente. Las versiones de copia de seguridad se mantienen en el subdirectorio `bak` del directorio del presente proyecto. Por defecto, JBuilder mantiene las diez versiones de copia de seguridad más recientes. Para cambiar el nivel de copia de seguridad (el número de versiones de copia de seguridad que mantiene JBuilder), seleccione Herramientas | Opciones del editor, abra la ficha Editor si no se había abierto automáticamente y ajuste el control de graduación Nivel de copia de seguridad. Se puede tener un máximo de 90 copias de seguridad almacenadas en el directorio `bak`.

9 Cambie `import java.applet.*;` a `import java.applet.Applet;` y guarde el archivo.

Nota El siguiente paso sólo es aplicable a usuarios de sistemas de control de versiones.

10 Si utiliza un sistema de control de versiones: confirme el archivo y escriba el comentario `Changed import.java.applet.* to import.java.Applet.`

Esto genera un tipo de control de versiones de la versión que se verá posteriormente.

11 Vaya a la línea 31, en la que se puede leer `boolean packFrame = false;`, y cambie `false` por `true`.

12 Guarde el archivo.

Nota Va a realizar otro cambio, pero *no guarde el archivo* hasta que se le indique.

13 Sitúe el cursor en la intersección de la línea 35 y la columna 23, donde se puede leer `WelcomeFrame frame = new WelcomeFrame();`, y cambie el nombre del objeto `frame` por `frame2`.

El código fuente tiene ahora este aspecto:

```
package com.borland.samples.welcome;

import java.awt.*;
import javax.swing.UIManager;
import java.applet.Applet;

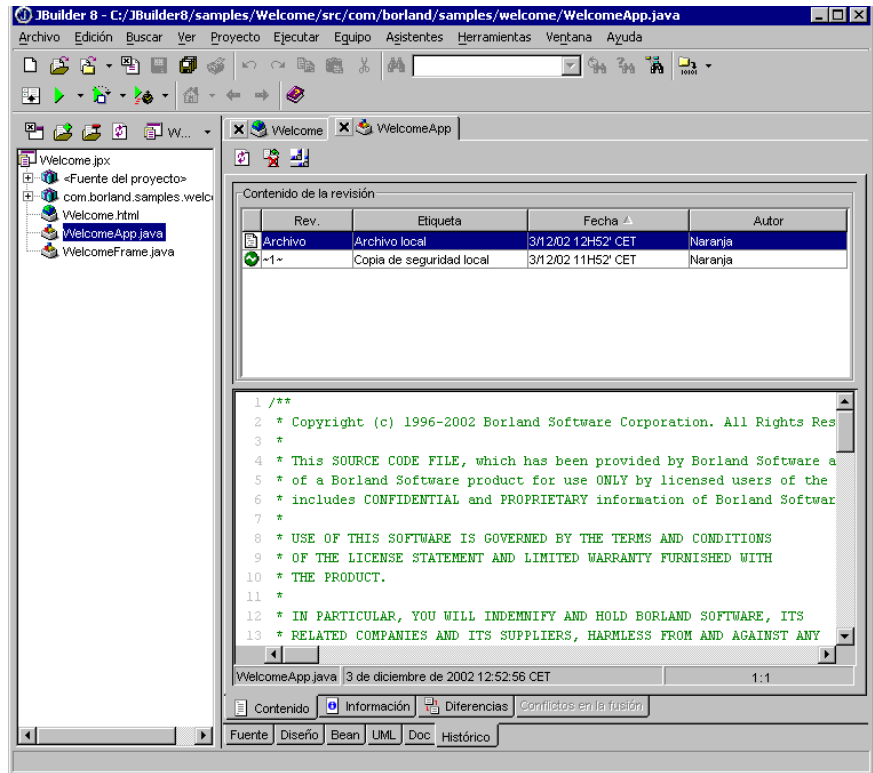
public class WelcomeApp {
    boolean packFrame = true;

    //Crear la aplicación
    public WelcomeApp() {
        WelcomeFrame frame2 = new WelcomeFrame();
    }
}
```

Paso 2: La ficha Contenido

La ficha Contenido muestra todas las revisiones del archivo seleccionado en el panel de contenido, permite ordenar los archivos de diversas formas y proporciona acceso a los botones Actualizar, Restaurar y Sincronizar desplazamiento.

1 Elija la pestaña de la vista Histórico. Se muestra la ficha Contenido:



Nota

La sección Contenido de la revisión, en la ficha Contenido, puede diferir de la ficha mostrada aquí, según se utilice o no el control de versiones.

La lista de revisiones de la parte superior presenta las versiones ordenadas por número de revisión, con las mas recientes en la parte superior de la lista.

El código fuente de la versión mas reciente se presenta en el visualizador de código fuente, debajo de la lista de revisiones.

Observe la información de la barra de estado: nombre del archivo, fecha y hora, y coordenadas de línea y columna.

- 2 Haga clic en la versión de un archivo para verla en el visualizador de código fuente.







JBuilder recupera la versión solicitada del lugar donde este almacenada, en el propio sistema de archivos o en el control de versiones.

- 3 Pulse el encabezado de la columna de iconos para ordenar las versiones por tipo de revisión.

Observe que se emplean iconos diferentes para los distintos tipos de versiones: copias de seguridad locales, control de versiones (si procede) y la última versión guardada:

Copia de seguridad local (tipo de versión universal)

Última versión guardada (tipo de versión universal)

Contenido de la revisión				
	Rev.	Etiqueta	Fecha ▲	Autor
	File	Local File	10/14/02 10:48:53 AM PDT	Administrador
	~3~	Local Backup	10/14/02 10:45:10 AM PDT	Administrador
	1.1.1.1	init	10/14/02 9:36:01 AM PDT	Administrador
	1.1		10/14/02 9:36:00 AM PDT	Administrador
	~2~	Local Backup	10/14/02 9:36:00 AM PDT	Administrador
	~1~	Local Backup	10/8/02 9:25:06 PM PDT	Administrador

Otras versiones VCS (tipo de versión VCS)

Versión obtenida del repositorio (tipo de versión VCS)

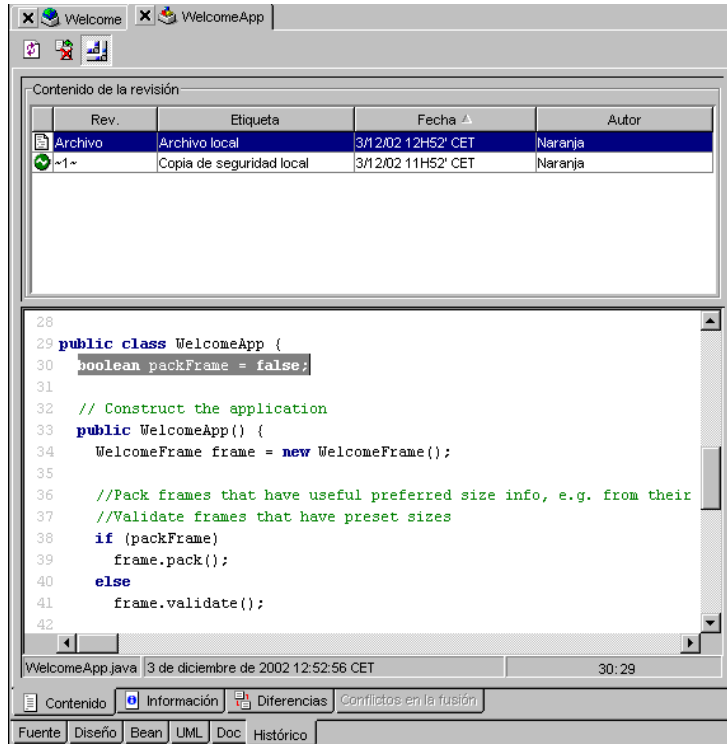
- 4 Pulse el encabezado de la columna Fecha para ordenar las versiones de forma descendente por fecha de modificación.
- 5 Seleccione el elemento superior de la lista (~1~ Local Backup).

Esta versión aparece en el panel de código fuente debajo de la lista de revisiones.



- 6 Pulse el botón Sincronizar desplazamiento.
- 7 Desplácese hasta la línea 30, que tiene la sentencia `boolean packFrame`.

8 Seleccione la sentencia completa y cópiela:



Elija Edición | Copiar o utilice el atajo de teclado. En la mayoría de las configuraciones de teclado, el método abreviado para copiar es *Ctrl + C*.

9 Seleccione la pestaña Fuente para volver al editor.

El texto que se presenta en el editor es el bloque que aparece en la ficha Contenido. Aunque la ventana de JBuilder sea pequeña, no es necesario desplazarse hacia abajo en el editor hasta encontrar la línea 31. Esto se debe a que se ha pulsado el botón Sincronizar desplazamiento.

10 En el editor, seleccione la línea 31 (la sentencia `boolean packFrame` en la última versión) y péguela en el texto copiado de la ficha Contenido de la vista Histórico.

Elija Edición | Copiar o utilice el atajo de teclado. En la mayoría de las configuraciones de teclado, el método abreviado para pegar es *Ctrl + V*.

11 Guarde el archivo.

La ficha Contenido de la ficha Histórico se puede utilizar para ver todas las versiones almacenadas de un archivo. Cuenta con los botones Restaurar y Sincronizar desplazamiento. Con el fin de proteger las versiones anteriores de los archivos, esta ficha las abre en modo de sólo lectura, y admite las operaciones de copiar y pegar texto de versiones anteriores al código fuente que se maneja actualmente.

Éste es el final del tutorial para los usuarios de JBuilder Personal. Para obtener más información sobre la vista Histórico y las fichas que contiene, consulte el [Capítulo 8, “Comparación de archivos y versiones”](#).

Paso 3: La ficha Diferencias

Es una función de JBuilder SE y Enterprise.

La ficha Diferencias muestra las diferencias entre dos versiones cualesquiera del archivo activo y proporciona acceso al botón Actualizar. Vamos a probarlo.

En este paso del tutorial aprenderá a:

- Ordenar revisiones en la sección Diferencias
- Deshacer cambios en una revisión seleccionada
- Utilizar Comparación inteligente para filtrar los cambios sin importancia al comparar versiones de archivos

Ordenación de las revisiones

Para ordenar la información de las revisiones:

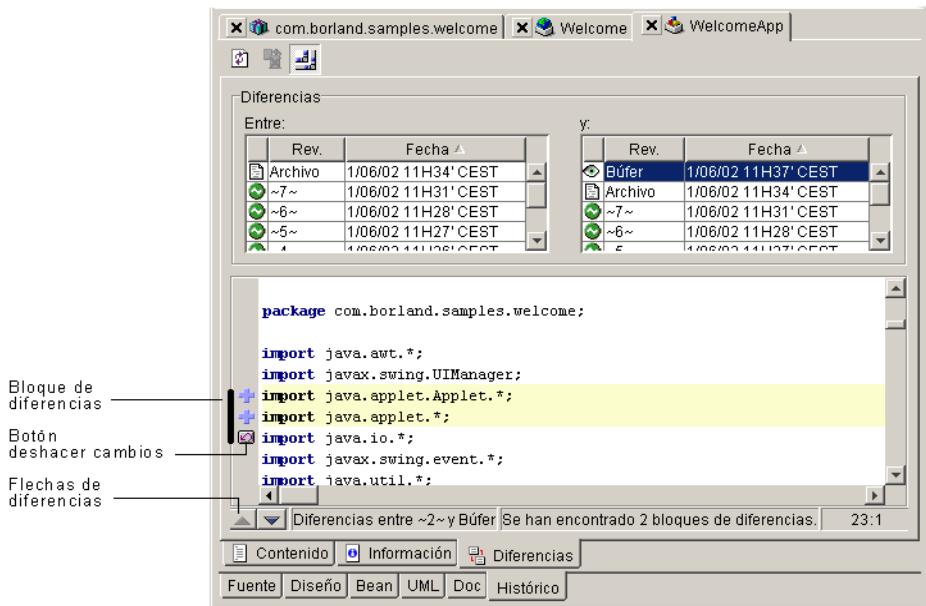
- 1 Seleccione la pestaña Diferencias en el panel histórico.
- 2 Pulse el encabezado Rev. del área Entre para ordenar las versiones por número de revisión.
- 3 Pulse el encabezado de la columna de iconos del área En si desea ordenar las versiones por tipos de gestión de revisiones.
Observe que las áreas Entre y En se ordenan de forma independiente.
- 4 Pulse el encabezado Rev. del área En.

Cómo deshacer cambios

Ahora va a ver las diferencias entre dos versiones de un archivo.

- 1 Seleccione ~2~ en el área Entre y Buffer en el área En.

JBuilder recupera las diferencias y las muestra en el visualizador de código fuente:



Si utiliza un sistema de control de versiones

En el área de estado, en la parte inferior, la ficha Diferencias indica el número de bloques de diferencias que hay en la comparación.

Para ver todos los cambios realizados en el repositorio desde que empezó a trabajar, seleccione la versión del repositorio original en el área De y la versión del archivo con el número de revisión más alto en el área Para.



2 Pulse el botón Actualizar.

Esta acción recupera las versiones nuevas del repositorio. No hay ninguna; por tanto, no cambia la lista de revisiones.

3 Para desplazarse de un bloque de código con diferencias a otro, pulse las flechas de diferencias de la izquierda de la barra de estado, situada en la parte inferior de la ficha Diferencias.

Como alternativa al uso de las flechas de diferencias para desplazarse entre los bloques, puede pulsar *Mayús+F10* para abrir un menú contextual (el cursor debe estar en el panel de la vista de código fuente) y seleccionar Anterior bloque de diferencias o Siguiente bloque de diferencias.

4 Pulse el botón Deshacer cambios.

Aparece una opción llamada Deshacer cambios anteriores.

- 5 Pulse esta opción. Ahora, en la línea 28 pone `import java.applet.*;`
`import java.applet.Applet;` desaparece y se resuelve el bloque
diferencias.

Comparación inteligente

La función Comparación inteligente permite excluir las diferencias debidas al formato del código y a cambios en los espacios en blanco cuando se comparan dos versiones de un archivo. Para observar cómo funciona Comparación inteligente:

- 1 Haga clic en la pestaña de la vista del archivo fuente.
- 2 Abra el cuadro de diálogo Propiedades de proyecto (Proyecto | Propiedades de proyecto) y elija la pestaña Formato.
- 3 Pulse Importar para abrir el cuadro de diálogo Importar configuración de formateo de código.
- 4 Seleccione el archivo de configuración `Expanded.codestyle` y pulse Aceptar.

El formato de código ampliado especificado por el archivo de configuración `Expanded.codestyle` aumenta significativamente la cantidad de espacios en blanco en los archivos.

- 5 Pulse Aceptar para cerrar el cuadro de diálogo Propiedades de proyecto.
- 6 Seleccione Edición | Formatear todo para aplicar los cambios de formato a `Wel comeApp.java`.

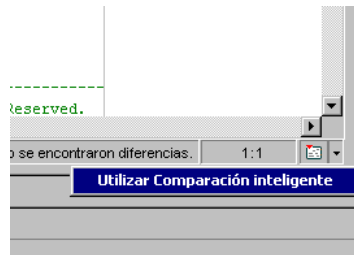
Observe cómo cambia el formato del código en el archivo.

- 7 Vuelva al panel histórico y seleccione la pestaña Diferencias.

Observe el incremento de bloques de diferencias debido al cambio de formato. Los bloques de diferencias adicionales se deben simplemente a las líneas y los espacios añadidos.



- 8 Pulse la flecha abajo situada junto al botón Comparación inteligente (ubicado en la esquina superior derecha de la ficha Diferencias) y seleccione la opción Utilizar Comparación inteligente.





El botón Comparación inteligente cambia, lo cual indica que la función de comparación inteligente está activada. Cuando se activa Comparación inteligente, todos los bloques, excepto el de las diferencias originales, desaparecen. La función de comparación inteligente puede resultar especialmente útil cuando varios desarrolladores comparten archivos y éstos presentan formatos adaptados a las preferencias de cada desarrollador, o bien se utiliza un formato global para guardar la coherencia.

La ficha Diferencias de la vista Histórico permite ver las diferencias entre todas las versiones disponibles de un archivo, ordenar las versiones de diversas formas, resolver las diferencias entre el búfer y otra versión, y desplazarse sin dificultad por las diferencias entre dos versiones.

Éste es el final del tutorial para los usuarios de JBuilder SE. Para obtener más información sobre la vista Histórico y las fichas que contiene, consulte el [Capítulo 8, “Comparación de archivos y versiones”](#).

Consulte

- [“Cómo aplicar formato al código” en la página 5-9](#) si desea obtener más información sobre la forma de especificar y aplicar preferencias de formato

Paso 4: La ficha Información

Es una función de
JBuilder Enterprise.

La ficha Información proporciona acceso a los comentarios y a las etiquetas de todas las revisiones de un archivo. Sus seis columnas se pueden utilizar como criterio de ordenación. Esta ficha también cuenta con los botones Restaurar y Actualizar.

- 1 Abra la pestaña Información.

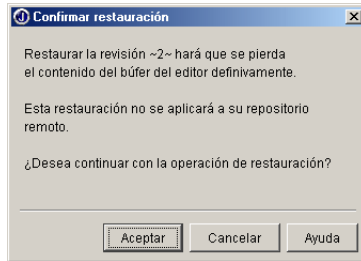
Debajo de la lista de revisiones aparecen la etiqueta y el comentario de la versión del archivo seleccionada en la lista.

- 2 Haga clic en una versión para ver su etiqueta y su comentario.
- 3 Pulse el encabezado de la columna de comentarios de la lista de versiones para ordenarla por comentarios.
- 4 Seleccione la versión número ~2~ de la copia de seguridad local.



5 Pulse el botón Restaurar.

Aparece el cuadro de diálogo Confirmar restauración.



Si se realiza la restauración desde una versión anterior, ésta pasa a ser la actual. Si el archivo está bajo el control de versiones, o si la capacidad de copia de seguridad es suficientemente alta, se conservan todas la versiones intermedias.

Si utiliza un sistema de control de versiones

El archivo restaurado se debe incorporar en el control de versiones, como cualquier otro cambio realizado en los archivos.

6 Pulse Aceptar.

Ha utilizado las fichas Contenido, Información y Diferencias de la vista Histórico. Ya sabe cómo ver diferentes revisiones de un archivo, buscar y ver sus comentarios y etiquetas, y examinar las diferencias entre las diversas versiones de copia de seguridad, búfer, repositorio y espacio de trabajo de los archivos.

Para obtener más información sobre la vista Histórico y las fichas que contiene, consulte el [Capítulo 8, “Comparación de archivos y versiones”](#).

Consulte

- El tutorial CVS en JBuilder de *Desarrollo en equipo con JBuilder* para ver un ejemplo de cómo utilizar la ficha Conflictos en la fusión

Índice

A

- Actualizar información de revisiones (botón del panel histórico) 8-5
- advertencia de sintaxis no válida 5-13
- ampliar el texto del código fuente 5-6
- aplicaciones
 - tutorial 3-1
- aplicar filtros a paquetes 4-7
- applets
 - tutorial 9-1
- árboles
 - buscar 4-18
 - desplazamiento 4-17
- archivos
 - comparar dos cualesquiera 8-1, 8-2
 - de proyecto
 - ver estructura 4-8
 - ver estructura 4-8
- área de trabajo
 - definición 8-1
- asistentes
 - galería de objetos 6-3
- aumentar el tamaño del texto del código 5-6
- ayuda
 - en pantalla 2-6
 - inmediata 5-17

B

- barra de división
 - Visualizador de aplicaciones 4-13
- barra de herramientas (ventana principal) 4-3
- barra de menús 4-3
- barras de estado
 - Visualizador de aplicaciones 4-16
- bloques de diferencias
 - definición 8-1
 - deshacer cambios 8-6
 - presentación 8-6
- Borland
 - asistencia
 - a desarrolladores 2-14
 - técnica 2-14
 - contacto 2-14
 - correo electrónico 2-16
 - grupos de noticias 2-15
 - informar sobre errores 2-16
 - recursos en línea 2-14
 - World Wide Web 2-15

- buscar 4-18
 - clases 4-18
 - con Buscar referencias 5-8
 - con escritura incremental 4-18
 - definiciones de símbolos 5-8
 - en árboles 4-18
 - en el editor 5-8
 - en el panel del proyecto 4-5
 - en el Visualizador de aplicaciones 4-17
 - referencias a símbolo de código 5-8
 - símbolo 5-17
 - texto 5-7
 - utilizar Buscar definición 5-8
- Buscar (menú) 5-7
- Buscar definición (comando) 5-8
- Buscar referencias (comando) 5-8

C

- carpeta Por hacer 4-10
- clases
 - buscar 4-18
- ClassInsight 5-15
- CodeInsight 5-13
 - configuración de teclado 5-14
 - configurar 5-18
 - error 5-14
- código
 - aplicar formato 5-10
 - automatizar 6-1
 - CodeInsight 5-13
 - completar automáticamente 5-14, 5-15
 - creación en el editor 5-1
 - dar formato 5-9
 - métodos abreviados 5-13
 - profundizar 4-10
- comandos
 - Buscar definición 5-8
 - Buscar referencias 5-8
 - de búsqueda 5-7
 - de método abreviado
 - codificación 5-13
 - de método abreviado para Javadoc 5-10
- comentarios
 - documentación de Javadoc 5-10
 - información de revisiones 8-8
- Comparación inteligente
 - activar 8-5
- Comparación inteligente (botón del panel histórico) 8-5

- comparar archivos
 - excluir cambios de formato 8-5
 - sincronizar desplazamiento 8-2
 - vista de diferencias 8-2
 - vista de fuente 8-2
- configuración del teclado 5-1
 - personalizar 5-2
- configurar
 - CodeInsight 5-18
- conflictos en la fusión
 - resolver 8-9
- Conflictos en la fusión (ficha) 8-9
- Contenido (ficha) 8-6
- control de versiones
 - comparar archivos 8-1
 - conflictos en la fusión 8-9
 - vistas
 - Histórico 8-4
- control de versiones (tipos)
 - iconos 8-4
- convenciones de la documentación 2-4, 2-5
- copiar
 - del visualizador de la ayuda 2-9
 - texto del panel de mensajes 4-15
- cuadros de diálogo
 - Opciones del IDE 4-20

D

- dar formato
 - aplicar al código 5-10
 - aplicar llaves a los bloques de código 5-10
 - código 5-9
- definición de teclado 4-16
- depurador 4-16
- depurar 5-17
- Depurar (ficha) 4-16
- desplazamiento
 - comandos de método abreviado 4-17
 - paneles del Visualizador de aplicaciones 4-17
 - visualizador de la ayuda 2-11
- diferencias
 - comparar dos archivos cualesquiera 8-2
 - definición 8-1
- Diferencias (ficha) 8-6
- diseñadores
 - de columnas 6-1
 - de interfaces de usuario 6-1
 - de menús 6-1
- documentación
 - convenciones 2-4
 - de plataformas 2-5
 - JBuilder 2-1
- documentación de Javadoc

- comandos de método abreviado 5-10
- comentarios 5-10
- documentación de referencia 2-12
 - diseño 2-13
- documentación de referencia de las bibliotecas 2-12
- documentación de referencia de las clases 2-12

E

- editor
 - buscar 5-7
 - comandos
 - Buscar definición 5-8
 - Buscar referencias 5-8
 - configuración del teclado 5-1
 - dividir la vista 5-3
 - etiquetas @todo 5-11
 - imprimir código fuente 5-12
 - mostrar números de línea 5-4
 - plantillas de código 5-18
 - seleccionar texto 5-5
 - sugerencias de Javadoc 5-10
 - utilizar 5-1
- emulaciones de editores 5-1
- enlaces rápidos
 - Favoritos 4-19
- errores de sintaxis
 - ver y buscar 4-11
- etiquetas
 - @todo 5-11
 - revisión 8-8
- examinar clases 4-18
- expresiones
 - evaluación automática 5-17
- ExpressionInsight 5-17

F

- Favoritos 4-19
- fecha
 - revisión 8-8
- fichas
 - Contenido 8-6
 - Depurar (cuadro de diálogo Propiedades de proyecto) 4-16
 - Diferencias 8-6
 - Punto de Observación 4-16
- flotante (panel de mensajes) 4-15
- fuentes
 - cambiar de tamaño en el editor 5-6
 - convenciones empleadas en la documentación de JBuilder 2-4

G

- galería de objetos 6-3
 - asistentes 6-3
- gestión de revisiones 8-4
- gestión de versiones
 - restaurar versiones de archivo anteriores 8-5
- glosario
 - términos de gestión de versiones 8-1
- grupos de noticias
 - Borland 2-15
 - public 2-15

H

- herramientas de diseño visual 6-1
- Histórico (vista) 8-4
 - Conflictos en la fusión (ficha) 8-9
 - descripción general 8-4
 - fichas
 - Contenido 8-6
 - Diferencias 8-6
 - iconos 8-4
 - Información (ficha) 8-8
 - tutorial 10-1

I

- iconos
 - control de versiones (tipos) 8-4
 - lista de revisiones 8-4
 - vista Histórico 8-4
- IDE
 - personalizar 4-20
- identificadores
 - comando Buscar definición 5-8
- impresión
 - código fuente 5-12
 - desde el editor 5-12
- Información (ficha) 8-8
- información de revisiones
 - presentación 8-8
- insertar clases
 - ClassInsight 5-15
- Ir a la línea 5-4

J

- Javadoc
 - carpeta Por hacer 4-10
 - etiquetas @todo 5-11
- JBuilder
 - Visualizador de aplicaciones 4-1

L

- libros
 - JBuilder 2-1
- lista de revisiones
 - actualizar 8-5
 - iconos 8-4
 - ordenar 8-6
- listas de parámetros 5-17
- llaves
 - colocar llaves en bloques de código 5-10
- lupa (herramienta) 4-16

M

- manuales
 - JBuilder 2-1
- marcadores
 - visualizador de la ayuda 2-10
- margen lateral
 - números de línea 5-4
 - utilizar para seleccionar texto 5-5
- MemberInsight 5-15
- mensajes de error
 - ver y buscar 4-11
- menús
 - atajos de barra de herramientas 4-3
 - comandos 4-3
- modo de asociación de teclas 4-16
- mostrar el panel de mensajes 4-15

N

- números de línea
 - ir a una línea determinada 5-4
 - mostrar en el editor 5-4
 - utilizar para seleccionar texto 5-5

O

- ocultar el panel de mensajes 4-15
- opciones de importación
 - ClassInsight 5-15
- opciones del IDE 4-20
- Opciones del IDE (cuadro de diálogo) 4-20
- ordenar
 - lista de revisiones 8-6

P

- paleta de componentes 6-1
- panel de contenido 4-12
 - ocultar otros paneles 4-13
 - pestañas de archivo 4-12
 - pestañas de vistas de archivo 4-13

- redimensionar 4-13
- vista Histórico 8-4
- panel de estructura 4-8
 - carpeta Por hacer 4-10
 - desplazarse por la estructura de archivo 4-8
 - mensajes de error 4-11
 - orden 4-8
 - usar para desplazarse por el código fuente 4-10
- panel de mensajes 4-13
 - copiar texto 4-15
 - integrar 4-15
 - mostrar 4-15
 - ocultar 4-15
 - segregar 4-15
- panel del proyecto
 - barra de herramientas proyecto 4-5
 - buscar 4-5
 - filtrar paquetes 4-7
 - menú contextual 4-7
- panel Fuente
 - dividir la vista 5-3
- ParameterInsight 5-17
- personalizar IDE 4-20
- pestañas
 - panel de contenido 4-12, 4-13
- plantillas de código 5-18
 - creación 5-18
- plataformas
 - convenciones 2-5
- profundizar
 - a la estructura de archivo 4-8, 4-10
- proyectos
 - añadir archivos 4-6
 - añadir clases 4-6
 - creación 4-6
- Punto de observación (ficha) 4-16

R

- reducir el tamaño del texto del código 5-6
- repositorio
 - definición 8-1
- Restaurar la revisión anterior (botón del panel histórico) 8-5
- revisión (etiqueta) 8-8
- revisión (fecha) 8-8
- revisiones (gestión) 8-4

S

- segregar panel de mensajes 4-15
- seleccionar
 - texto desde el margen lateral del número de línea 5-5

- símbolos de código
 - buscar definición 5-8
 - buscar referencias a 5-8
- sincronizar desplazamiento
 - en visualizadores de código fuente 8-5
- Sincronizar desplazamiento (botón del panel histórico) 8-5

T

- tamaño de fuente
 - cambiar en el editor 4-16
- texto
 - buscar en la ayuda 2-8
 - búsqueda completa 2-8
 - tema actual 2-9
 - localizar 5-7
 - seleccionar en el editor 5-5
- texto del código fuente (cambiar tamaño) 5-6
- tutoriales
 - creación de aplicaciones 3-1
 - creación de un applet 9-1
 - utilizar la vista del histórico 10-1

U

- Usenet, grupos de noticias 2-15

V

- versiones de archivo anteriores
 - restaurar 8-5
- versiones previas (ver) 8-4
- vistas
 - Histórico 8-4, 8-6, 8-8, 8-9, 10-1
 - jerárquicas 4-8
- Visualizador de aplicaciones 4-1
 - abrir varios visualizadores de aplicaciones 4-2
 - ampliar árboles 4-17
 - barra de herramientas 4-3
 - barra de menús 4-3
 - barras de estado 4-16
 - buscar 4-17
 - contraer árboles 4-17
 - depurador 4-16
 - desplazarse por los paneles
 - desplazamiento
 - árboles del Visualizador de aplicaciones 4-17
 - editor 5-1
 - ocultar paneles 4-13
 - panel de contenido 4-12
 - panel de estructura 4-8
 - panel de mensajes 4-13
 - panel del proyecto 4-5

- redimensionar panel de contenido 4-13
- visualizador de la ayuda
 - ampliar la vista 2-9
 - búsqueda completa de texto 2-8
 - búsqueda de texto en tema actual 2-9
 - colocar marcadores 2-10
 - copiar texto 2-9
 - desplazamiento 2-11
 - desplazarse con el teclado 2-12

- índice alfabético 2-8
- índice de materias 2-8
- interfaz de usuario 2-7
- utilizar 2-8
- visualizadores de código fuente
 - sincronizar desplazamiento 8-5

