

Introducción a JBuilder®

JBuilder® X

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

Consulte el archivo `deploy.html` ubicado en el directorio `redist` de su producto JBuilder para acceder a una lista completa de archivos que puede distribuir de acuerdo con su Declaración de licencia y Garantía limitada de JBuilder.

Borland Software Corporation puede tener patentes y/o solicitudes de patente pendientes en lo relativo a los asuntos tratados en este documento. Consulte el CD del producto o el cuadro de diálogo Acerca de para ver la lista de patentes aplicables. La entrega de este documento no supone la cesión de ninguna licencia sobre estas patentes.

Copyright © 1997–2004 Borland Software Corporation. Reservados todos los derechos. Todas las marcas y nombres de productos Borland son marcas comerciales o registradas de Borland Software Corporation en los Estados Unidos y en otros países. Las demás marcas pertenecen a sus respectivos propietarios.

Para conocer las condiciones y limitaciones de responsabilidad de terceros, consulte las Notas de la versión en el CD del product Borland.

Impreso en los EEUU.

JXE0010WW21000intobjb 5E4R1103
0304050607-9 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 visor de ayuda	2-7
El visor de ayuda	2-8
El índice de materias	2-9
El índice alfabético	2-9
Búsqueda completa de texto (incluye la búsqueda booleana)	2-9
Búsqueda de texto en el tema actual	2-10
Copia de texto del visor de ayuda	2-11
Ampliación en el visor de ayuda	2-11
Utilización de marcadores	2-11
Configuración de los parámetros de proxy	2-12
Inicio del visor de ayuda en una MV independiente	2-12
Desplazamiento por el visor de ayuda	2-13
Desplazamiento mediante el teclado	2-14
Cambio entre los paneles del visor de ayuda	2-14
Añadición de pestañas al panel de contenido del Visor de ayuda	2-14
Consulta de la documentación de referencia de las clases	2-15
Diseño de la documentación de referencia	2-15
Impresión de tutoriales y otros tipos de documentación de JBuilder	2-16
Técnicas adaptativas aceptadas por JBuilder	2-16
Cumplimiento de la sección 508	2-17
Recursos adicionales para obtener más información sobre JBuilder	2-17
Asistencia y recursos para desarrolladores	2-17
El servicio de asistencia técnica de Borland	2-17
Recursos en línea	2-18
World Wide Web	2-18
Grupos de noticias de Borland	2-18

Usenet, grupos de noticias 2-19

Información sobre errores 2-19

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	3-8
Paso 4: Personalización de la interfaz de usuario del applet	3-9
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	3-17
Paso 9: Asignación de sucesos a botones	3-18
Paso 10: 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 el archivo JAR	3-23
Paso 13: 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 HolaATodosClase.java	3-27
Código fuente de HolaATodosMarco_AcercaDe.java	3-28

Capítulo 4

El IDE de JBuilder

4-1

Menú principal	4-3
Barra de herramientas principal	4-4
Configuración del espacio de trabajo	4-6
La configuración "clásica" de JBuilder	4-7
Gestión de las configuraciones	4-8
Modificación de las configuraciones	4-8
Maximización y restablecimiento de los paneles	4-8
Iconización, integración y segregación de paneles	4-9
Presentación de varios archivos	4-10

Panel del proyecto	4-11
La pestaña de vista Proyecto del panel del proyecto	4-11
La pestaña de vista Buscador de archivos del panel del proyecto	4-12
Ejecución de aplicaciones desde el panel del proyecto	4-12
Creación de proyectos	4-13
Adición de archivos, clases, paquetes o aplicaciones a un proyecto	4-13
Creación de archivos	4-13
Adición de archivos vacíos	4-14
Adición de aplicaciones	4-14
Adición de archivos existentes	4-14
Adición de archivos, carpetas, directorios o paquetes	4-14
Creación de JAR ejecutables para aplicaciones	4-16
Creación de archivos y paquetes para aplicaciones web	4-16
El panel del proyecto	4-16
Panel de estructura	4-17
Ordenación en el panel de estructura	4-17
Filtrado en el panel de estructura	4-19
Presentación de la información de Javadoc en el panel de estructura	4-20
El panel de estructura permite desplazarse por el código fuente	4-20
Presentación de los mensajes de error en el panel de estructura	4-21
Panel de contenido	4-22
Pestañas de archivos del panel de contenido	4-22
Pestañas de vista de archivos del panel de contenido	4-23
Redimensionamiento del panel de contenido	4-24
Panel de mensajes	4-24
Copia de texto del panel de mensajes	4-25
Ocultación y presentación del panel de mensajes	4-26
Segregación del panel de mensajes	4-26
Depurador	4-26
Barras de estado	4-27
Desplazamiento y búsqueda en el IDE	4-27
Búsqueda en árboles	4-28
Búsqueda de clases	4-29
Utilización de Favoritos	4-30
Personalización del IDE	4-32

Capítulo 5

El editor

5-1

Personalización del editor	5-3
Selección de una configuración de teclado para el editor	5-3
Modificación de las configuraciones de teclado	5-4
Creación de copias de referencia de las configuraciones de teclado	5-5
División de la vista del código fuente	5-6
Presentación de los números de línea	5-7
Control de los números de línea	5-7
Desplazamiento a una línea determinada	5-8
Selección, formato y búsqueda de texto	5-8
Selección de texto	5-8
Desplazamiento de texto mediante la operación de arrastrar y soltar	5-9
Cambio del tamaño de fuente	5-9
Cómo aplicar formato al código	5-10
Acceso a la configuración de formato	5-10
Aplicación de formato al código fuente	5-11
Personalización de elementos de pantalla	5-12
Colocación de llaves en los bloques de código	5-13
Búsqueda de texto en el editor	5-13
Búsqueda de elementos de código y definiciones	5-14
Búsqueda de definiciones de símbolos	5-14
Búsqueda de referencias a símbolos	5-15
Búsqueda del método redefinido	5-15
Búsqueda de elementos de código y métodos que redefinen a otros	5-15
Aplicación de accesos directos de Javadoc	5-16
Las plantillas de código de Javadoc	5-16
Adición y modificación de etiquetas Javadoc	5-17
JavadocInsight	5-18
Resolución de conflictos Javadoc	5-19
Utilización de etiquetas @todo en el editor	5-19
Visualización de etiquetas @todo Javadoc	5-20
Definición de marcadores	5-20
Adición de marcadores	5-21
Modificación de marcadores	5-21
Presentación de marcadores	5-21
Desplazamiento a los marcadores	5-22
Desplazamiento, apertura y adición de archivos	5-22

Métodos abreviados de codificación	5-22
Compatibilidad de impresión en el editor	5-23
Capítulo 6	
Automatización del desarrollo de código	6-1
Las herramientas de diseño visual	6-2
La Galería de objetos	6-3
Inicio rápido por medio de los asistentes	6-4
Creación de archivos JAR ejecutables con el asistente Creador de recopilatorios	6-4
Creación de elementos de código.	6-5
Los métodos abreviados de codificación	6-6
CodeInsight	6-7
MemberInsight	6-9
ClassInsight.	6-9
ParameterInsight	6-10
Evaluación inmediata de expresiones	6-11
ExpressionInsight.	6-11
Configuración de CodeInsight	6-11
ScopeInsight	6-11
Ocultación de código.	6-13
Desplazamiento a miembros de clase	6-14
ErrorInsight.	6-14
Plantillas de código	6-17
Adición de plantillas al código mediante la escritura de su nombre.	6-18
Adición de plantillas al código mediante el cursor.	6-18
Adición de tipos de archivos para plantillas de código	6-18
Adición de sentencias de importación al código mediante la importación automática.	6-19
Adición de macros de plantilla a las plantillas de código	6-19
Creación de plantillas de código	6-20
Creación de plantillas de código a partir del código fuente	6-20
Inserción del código fuente en plantillas de código	6-21
Importación y exportación de plantillas de código	6-21
Modificación de plantillas de código	6-22
Desplazamiento mediante pestañas por el bloque de plantilla de código insertado	6-23
Sync Edit.	6-23
Edición con Sync Edit.	6-23

Personalización de los elementos de codificación de Sync Edit	6-25
TagInsight	6-25
EntityInsight	6-27
XML TagInsight.	6-27
Personalización de TagInsight	6-28

Capítulo 7	
Proyectos de JBuilder	7-1
Creación de proyectos	7-2
Adición de archivos a un proyecto	7-4
Presentación de archivos	7-7
Adición de un archivo fuente Java	7-7
Eliminación y borrado de archivos	7-10
Cómo cambiar de nombre proyectos y archivos	7-10
Apertura de proyectos	7-11
Cierre de proyectos	7-11
Más información sobre los proyectos de JBuilder	7-11

Capítulo 8	
Tutorial: Creación de un applet	8-1
Aspectos generales.	8-2
El applet Buenas noches	8-2
Paso 1: Creación de un proyecto	8-3
Paso 2: Generación de los archivos de código fuente	8-7
Paso 3: Compilación y ejecución del applet	8-12
Paso 4: Personalización de la interfaz de usuario del applet	8-14
Paso 5: Incorporación de componentes AWT al applet	8-19
Paso 6: Modificación del código fuente	8-22
Paso 7: Distribución del applet	8-27
Distribución del applet con el Creador de recopilatorios	8-28
Paso 8: Modificación del archivo HTML	8-32
Paso 9: Ejecución del applet distribuido desde la línea de comandos	8-33
Comprobación del applet distribuido en la Web.	8-35
Código fuente del applet	8-35
Código fuente HTML del applet BuenasNochesClass.html	8-36
Código fuente de la clase del applet BuenasNochesClase.java	8-36

Índice	I-1
---------------	------------

Tablas

2.1	Convenciones tipográficas y de símbolos.	2-4	5.2	Variables del encabezado	5-23
2.2	Convenciones de plataformas:	2-5	6.1	Comandos del submenú Nuevo (menú contextual del editor)	6-6
4.1	Iconos de la barra de herramientas principal	4-4	6.2	Combinaciones de teclas	6-8
4.2	Filtros del panel de estructura.	4-19	6.3	Opciones del menú ErrorInsight	6-15
4.3	Configuraciones de teclado para desplazamientos:	4-27			
5.1	Comandos de búsqueda de texto . . .	5-13			

Figuras

2.1	Visor de ayuda de JBuilder	2-7	6.1	JBuilder en vista diseño.	6-2
3.1	Los elementos del IDE de JBuilder . . .	3-8	6.2	La Galería de objetos.	6-3
3.2	Elementos del diseñador de interfaces de usuario	3-10	6.3	Ventana emergente de CodeInsight: . .	6-8
4.1	La ventana del IDE	4-2	6.4	Ventana emergente de ElementInsight	6-26
4.2	Menú contextual del panel de proyecto	4-15	6.5	Ventana emergente de EntityInsight .	6-27
4.3	Menú contextual del panel de estructura	4-18	8.1	Diseñador de interfaces de usuario . .	8-14
4.4	La ficha Estructura Java.	4-19			
4.5	Panel de contenido que muestra la vista fuente (ofrece un editor).	4-22			
4.6	Ficha Buscar	4-29			
4.7	Ficha Examinar:	4-30			
5.1	Editor de configuración de teclado CUA	5-5			

Tutoriales

Creación de una aplicación	3-1	Creación de un applet	8-1
--------------------------------------	-----	---------------------------------	-----

Introducción

Introducción a JBuilder es el punto de partida para los usuarios que no conocen JBuilder. Es una introducción de los recursos disponibles para ayudarle a aprender acerca de JBuilder, muestra cómo debe utilizar el Visor de ayuda en línea y le pone rápidamente manos a la obra con un tutorial que crea su primera aplicación con JBuilder. Tras esta primera experiencia con el entorno de desarrollo de JBuilder, también denominado buscador, podrá familiarizarse con el editor, el diseñador, y con todas las funciones de productividad que le ayudarán a crear el código fuente. Se presenta el aspecto de los diversos paneles, pestañas, barras de herramientas, iconos y menús del IDE de JBuilder, y se muestra cómo utilizarlos para realizar tareas habituales.

También se enseña la forma de utilizar el editor de JBuilder, que incluye varias herramientas y métodos abreviados para escribir código. Se familiarizará con las herramientas de diseño visual para crear rápidamente IU y con los asistentes para generar código automáticamente. También aprenderá a crear y gestionar proyectos de JBuilder. Por último, tendrá la oportunidad de conocer otro tutorial básico 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, lo que incluye la documentación y la ayuda en pantalla.

- [Capítulo 3, “Tutorial: Creación de una aplicación”](#)

Explica cómo crear una sencilla aplicación “Hola a todos” con JBuilder.

- [Capítulo 4, “El IDE de JBuilder”](#)

Presenta el entorno de desarrollo integrado de JBuilder y describe la forma de utilizar sus menús, barras de herramientas, paneles, herramientas, barras de estado, navegación y espacios de trabajo.

- [Capítulo 5, “El editor”](#)

Explica cómo se utilizan las funciones del editor de código de JBuilder, incluidos los marcadores, los métodos abreviados de teclado, el resaltado de sintaxis, la correspondencia de llaves, la búsqueda y el desplazamiento por el código, los comentarios Javadoc y los métodos abreviados de escritura de código.

- [Capítulo 6, “Automatización del desarrollo de código”](#)

Describe cómo se genera el código automáticamente mediante las herramientas de diseño visual, los asistentes de JBuilder y los métodos abreviados de codificación, incluidos CodeInsight, JavadocInsight y plantillas de código.

- [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, “Tutorial: Creación de un applet”](#)

Explica cómo crear un applet AWT.

Cómo obtener información sobre JBuilder

JBuilder proporciona numerosos recursos de documentación que ayudan a los usuarios a familiarizarse con sus funciones y sus herramientas. En este apartado se describe el amplio conjunto de documentación de JBuilder y se explica la forma de utilizar el sistema de ayuda en pantalla de JBuilder. El apartado de ayuda en pantalla describe los principales componentes del Visor de ayuda y la forma de utilizarlos, así como la forma de consultar la documentación de las referencias de clase. Familiarícese con las herramientas del Visor de ayuda, que incluyen herramientas de navegación, búsqueda, de las pestañas y de los marcadores para consultar la documentación con más facilidad. Lea este apartado para aprovechar al máximo las funciones que ofrece el sistema de ayuda.

Aquí se explican también la impresión de la documentación, el desplazamiento mediante el teclado, las opciones de presentación y otras técnicas adaptativas aceptadas por JBuilder. También se enumeran otros lugares donde se puede encontrar información adicional sobre el uso de JBuilder.

Documentación de JBuilder

Se puede acceder de varias formas a la documentación de JBuilder. Puede ver la documentación en el visor de ayuda de JBuilder mediante las opciones de menú AyudaTemas de ayuda, AyudaEntorno de JBuilder o Ayuda

Tutoriales de JBuilder. Para más información sobre ver la ayuda en el Visualizador de ayuda, consulte [“La ayuda en pantalla de JBuilder” en la página 2-6](#). 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, en <http://www.borland.com/techpubs/jbuilder/>

Documento	Descripción
Todas las ediciones de JBuilder	
<i>Introducción a JBuilder</i>	Indica dónde puede encontrar la información que desea obtener en la documentación de JBuilder y muestra cómo se utiliza el visor de ayuda. Realiza una rápida introducción a la programación mediante un tutorial que guía en la creación de una aplicación sencilla de Java con JBuilder. Presenta el IDE de JBuilder, el editor de código y algunos de los muchos asistentes y herramientas de codificación 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 otro tutorial sobre la creación de un applet sencilla con JBuilder.
<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 aplicaciones 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.
<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.

Documento	Descripción
<i>Desarrollo de aplicaciones de base 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>Referencia de la biblioteca de componentes DataExpress</i>	Contiene información detallada sobre todos los componentes enlazados a datos, clases, propiedades, métodos y sucesos de valor añadido de JBuilder.
<i>Referencia de la API de OpenTools</i>	Proporciona documentación de referencia sobre la API de OpenTools de JBuilder.
<i>Documentación de JDK</i>	Documentación de referencia de la API para el Kit de desarrollo de Java (JDK) de Sun Se puede acceder a esta documentación de dos formas: <ul style="list-style-type: none"> ■ Elija AyudaReferencia de Java ■ Seleccione la pestaña Doc del panel de contenido con un archivo .JDK abierto
<i>Especificación del lenguaje Java</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 Developer y 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 XML</i>	Trata la tecnología que se utiliza para el desarrollo en XML con JBuilder.
<i>Desarrollo de aplicaciones para dispositivos móviles para MIDP</i>	Explica cómo utilizar el entorno de desarrollo integrado de JBuilder para crear y recopilar aplicaciones MIDP para dispositivos móviles preparados para J2ME. Muestra cómo gestionar proyectos y archivos, diseñar visualmente una interfaz de usuario y compilar, ejecutar, depurar y recopilar aplicaciones MIDP.
<i>Desarrollo de aplicaciones para dispositivos móviles para i-mode</i>	Explica cómo utilizar el entorno de desarrollo integrado de JBuilder para crear y recopilar aplicaciones J2ME para dispositivos móviles habilitados para i-mode™ con el SDK de DoJa. Muestra cómo gestionar proyectos y archivos, diseñar visualmente una interfaz de usuario y compilar, ejecutar, depurar y recopilar aplicaciones Java i-mode.
Edición Enterprise	
<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>Desarrollo de aplicaciones para servidores J2EE</i>	Presenta una descripción general del desarrollo de J2EE y explica cómo utilizar los servidores J2EE con JBuilder.
<i>Desarrollo de aplicaciones con 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

Tipografía	Significado
Negrita	La negrita se utiliza para las herramientas java, <code>bmj</code> (Borland Make for Java-Make de Borland para Java), <code>bcj</code> (Borland Compiler for Java - El Compilador Borland para Java) y opciones del compilador. Por ejemplo: <code>javac</code> , <code>bmj</code> , <code>-classpath</code> .
<i>Cursiva</i>	Las palabras en cursiva indican los términos nuevos que se definen y los títulos de libros; ocasionalmente se usan para indicar énfasis.
<i>Teclas</i>	Las teclas, como “Pulse <i>Esc</i> para salir de un menú”.
Letra monoespaciada	El tipo monoespaciado representa lo siguiente: <ul style="list-style-type: none"> ■ texto tal y como aparece en la pantalla ■ texto que el usuario debe introducir, como en “Escriba <code>Hola a todos</code> en el campo Título del Asistente para aplicaciones” ■ nombres de archivos ■ nombres de vías de acceso ■ nombres de directorios y carpetas ■ comandos, como <code>SET PATH</code> ■ 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>
[]	Los corchetes, en las listas de texto o sintaxis, encierran elementos optativos. En estos casos no se deben escribir los corchetes.

Tabla 2.1 Convenciones tipográficas y de símbolos (continuación)

Tipografía	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 empleado.jds, 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: o <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 de texto se utiliza para indicar cadenas de variables dentro de los ejemplos de código que ya utilizan corchetes angulares como delimitadores. Por ejemplo,</p> <pre><url="jdbc:borland:jbuilder\samples\guestbook.jds"></pre>
...	<p>En los ejemplos de código, los puntos suspensivos (...) indican código que se ha omitido para ahorrar espacio y mejorar 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. En la tabla siguiente se proporciona una descripción de las convenciones de plataformas utilizadas en la documentación.

Tabla 2.2 Convenciones de plataformas:

Elemento	Significado
Vías de acceso	<p>En las vías de acceso de la documentación se utiliza la barra normal (/).</p> <p>En la plataforma Windows se utiliza la barra invertida (\).</p>
Directorio de inicio	<p>La ubicación del directorio inicial varía según la plataforma y se indica con la 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	<p>Las imágenes o capturas de pantalla utilizan el aspecto Metal en diversas plataformas.</p>

La ayuda en pantalla de JBuilder

JBuilder presenta los temas de ayuda en pantalla en el visor de ayuda. Los temas de ayuda también se pueden consultar en el IDE de JBuilder y en un navegador web.

Si desea información más detallada sobre el sistema de ayuda de JBuilder, elija uno de los siguientes temas:

- “Obtención de ayuda” en la página 2-6
- “Principales partes del visor de ayuda” en la página 2-7
- “El visor de ayuda” en la página 2-8
- “Desplazamiento por el visor de ayuda” en la página 2-13
- “Consulta de la documentación de referencia de las clases” en la página 2-15
- “Impresión de tutoriales y otros tipos de documentación de JBuilder” en la página 2-16

Obtención de ayuda

Existen varias formas de obtener ayuda sobre cualquier tema:

- Si desea obtener ayuda desde el IDE:
 - Para abrir el visor de ayuda, elija Temas de ayuda en el menú Ayuda de JBuilder.
 - Para abrir el Visor de ayuda, pulse el botón Temas de ayuda de la barra de herramientas principal de JBuilder.
 - Pulse el botón Ayuda que aparece en los cuadros de diálogo de JBuilder 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.
 - Sitúe el cursor sobre una palabra clave de Java, en el editor, y pulse *F1* para obtener ayuda.
 - Haga doble clic en un nombre de clase Java, en el panel de estructura o del proyecto, y seleccione la pestaña **Doc**. Aparece la documentación de esta clase.
 - Seleccione un nombre de clase Java del panel de estructura o el del proyecto y pulse *Intro*; a continuación, haga clic en la pestaña **Doc**. Equivale a hacer doble clic.
- 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:

```
<JBuilder>/<jdk>/bin/jar xf <docjarfile.jar>
```

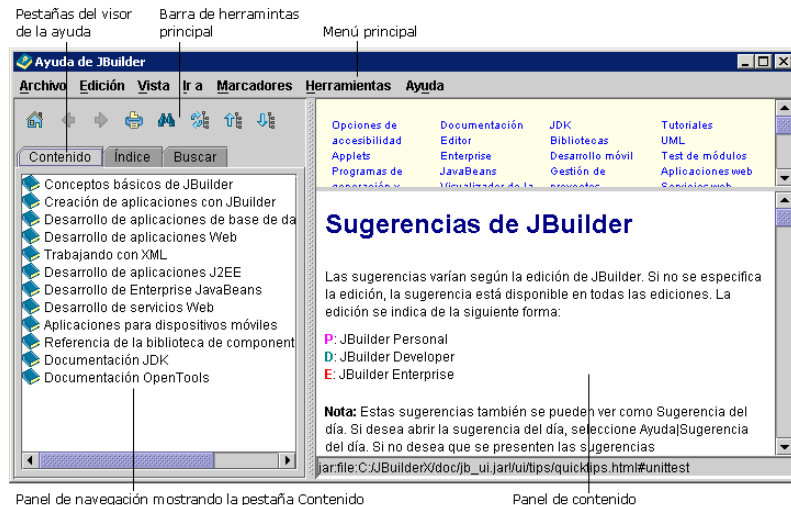
- b** Después de extraer todos los archivos JAR, abra `index.html` con el navegador y siga las instrucciones.

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 visor de ayuda

El Visor de ayuda de JBuilder contiene cómodas funciones para buscar y mostrar la documentación de ayuda. Esto incluye el menú principal, la barra de herramientas principal y los paneles de navegación y contenido.









Figura 2.1 Visor de ayuda de JBuilder



El visor de ayuda de JBuilder incluye lo siguiente:

- El menú principal.
- El panel de navegación, que contiene las siguientes ficha a las que se accede desde las pestañas del Visor de ayuda:
 - La ficha Contenido, que muestra una tabla de contenido para todos los manuales en una vista arbolada.
 - 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.

- La barra de herramientas principal, que cuenta con 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.
	Sincronizar	Sincroniza la selección de temas de la ficha Contenido con el tema que se ve en el panel de contenido. Si está leyendo un tema y desea ver a qué parte del sistema de ayuda pertenece, pulse este botón y se abrirá la tabla de contenido, con un esquema del camino que conduce al archivo abierto.
	Tema anterior	Abre el tema anterior en la tabla de contenidos.
	Tema siguiente	Abre el tema siguiente en la tabla de contenidos.

El visor de ayuda

Si desea información sobre la utilización del visor de ayuda, elija uno de los siguientes temas:

- [“El índice de materias” en la página 2-9](#)
- [“El índice alfabético” en la página 2-9](#)
- [“Búsqueda completa de texto \(incluye la búsqueda booleana\)” en la página 2-9](#)
- [“Búsqueda de texto en el tema actual” en la página 2-10](#)
- [“Copia de texto del visor de ayuda” en la página 2-11](#)
- [“Ampliación en el visor de ayuda” en la página 2-11](#)
- [“Utilización de marcadores” en la página 2-11](#)
- [“Configuración de los parámetros de proxy” en la página 2-12](#)
- [“Inicio del visor de ayuda en una MV independiente” en la página 2-12](#)

El índice de materias

Para elegir un tema de ayuda del índice de materias:

- 1 Haga clic en la pestaña Contenido, en la esquina superior izquierda del visor 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.

Cuando se utiliza la pestaña Buscar o Índice del Visor de ayuda, la tabla de contenido visible en la pestaña Contenido no está disponible. Sin embargo, cuando tenga la documentación podrá encontrar fácilmente la ubicación de los documentos deseados:

- 1 Busque el documento con la pestaña Buscar o Índice.
- 2 Con el documento abierto en el Visor de ayuda, pulse el botón Sincronizar tabla de contenidos de la barra de herramientas Ayuda.

La tabla de contenidos se amplía, con el encabezado del tema resaltado.



El índice alfabético

Para buscar un tema en el índice:

- 1 Haga clic en la pestaña Índice, situada en la parte superior izquierda del visor 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)

Si desea utilizar las funciones de búsqueda de la ficha Buscar para buscar texto en todos los libros:

- 1 Haga clic en la pestaña Buscar, situada en la esquina superior izquierda del visor de ayuda.
- 2 Escriba las palabras para buscar en el campo de texto y pulse *Intro* para ejecutar la búsqueda.

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

Nota

La búsqueda de texto completa no distingue entre mayúsculas y minúsculas.

El visor de ayuda admite las siguientes operaciones de búsqueda avanzada:

Operación	Operadores	Descripción
Intersección de temas	Signo más (+) o barra espaciadora	<p>Escriba un signo más (+) o un espacio entre las palabras si desea buscar los temas que incluyan todas esas palabras.</p> <p>Por ejemplo, si escribe <code>JBuilder Help System</code> en el campo Buscar, aparecen los temas que incluyen las tres palabras, en cualquier orden.</p>
Unión de temas	Coma (,) o barra vertical ()	<p>Escriba una coma (,) o una barra vertical () entre las palabras si desea buscar todos los temas que incluyan al menos una de esas palabras.</p> <p>Por ejemplo, si escribe <code>Borland,Sun</code> en el campo Buscar, aparecen todos los temas que contienen Borland o Sun.</p>
Exclusión de temas	Signo menos (-)	<p>Escriba un signo menos (-) delante de las palabras para excluir de los resultados de búsqueda los temas que contengan esas palabras.</p> <p>Por ejemplo, si escribe <code>JDK-Sun</code> en el campo Buscar, aparecen los temas que contienen JDK pero no Sun.</p>
Frase exacta	Comillas (" ")	<p>Utilice comillas (" ") en una frase para buscar temas que contengan la frase exacta.</p> <p>Por ejemplo, si escribe <code>"What's new"</code> en el campo Buscar, aparecen los temas con esta frase exacta.</p>
Operaciones anidadas	Paréntesis (())	<p>Utilice paréntesis (()) para construir una búsqueda booleana compleja.</p> <p>Por ejemplo, si escribe <code>recopilatorio-(jar war)</code> en el campo Buscar, aparecen todos los temas que incluyen recopilatorio, pero no jar ni war.</p>

Si se encuentra más de un tema para una búsqueda, la ficha Buscar abre un panel Temas encontrados en la parte inferior de la ficha. El panel Temas encontrados recoge todos los temas que coinciden con los criterios de búsqueda del campo Buscar. Pulse el tema deseado.

Sugerencia

La búsqueda completa de texto devuelve resultados de todo el texto en todos los manuales de la documentación en línea. 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 visor de ayuda.



- 1 Pulse el botón Buscar en el visor de ayuda o *Ctrl+F*.
- 2 Escriba las palabras en el cuadro de diálogo Buscar en tema actual.
- 3 Pulse Buscar siguiente o *Intro* para continuar con la búsqueda.

- 4 Haga clic en Cerrar para cerrar el cuadro de diálogo.
- 5 Pulse *F3* para buscar de nuevo sin abrir el cuadro de diálogo.

Copia de texto del visor de ayuda

Para copiar texto del visor de ayuda:

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

Ampliación en el visor de ayuda

Es posible ampliar y reducir el tamaño de la fuente y las imágenes que se muestran en el visor 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+TecNum+* (signo más del teclado numérico [+], a la derecha del teclado; mantenga pulsada la tecla *Ctrl* y vuelva a pulsar + en el teclado numérico hasta que la vista tenga el tamaño deseado.
- Seleccione Ver|Ampliar.

Para reducir:

- Pulse *Ctrl+TecNum-* (signo menos del teclado numérico [-], a la derecha del teclado; mantenga pulsada la tecla *Ctrl* y vuelva a pulsar - en el teclado numérico hasta que la vista tenga el tamaño deseado.
- Seleccione Ver|Reducir.

Para volver a la vista original:

- Pulse *Ctrl+TecNum** (asterisco del teclado numérico [*], a la derecha del teclado).
- Seleccione Ver|Normal.

Utilización de marcadores

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

Para modificar marcadores en el visor de ayuda:

- 1 Seleccione Marcadores|Modificar marcador 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.

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 50 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 visor 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 visor 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 visor de ayuda.

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

- 1 Abra el visor de ayuda (Ayuda|Temas de ayuda).
- 2 Elija Herramientas|Preferencias en el menú principal del Visor de ayuda 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 Active o desactive la opción Utilizar cuadro de diálogo sencillo para la ayuda sensible al contexto.
 - Activada (configuración por defecto): Abre un cuadro de diálogo sencillo que muestra información de ayuda cuando se pulsa *F1* en JBuilder.
 - Desactivada: Abre un toda la interfaz del Visor de ayuda, con información de ayuda, cuando se pulsa *F1* en JBuilder.
- 7 Pulse Aceptar.
- 8 Reinicie JBuilder para que se aplique el cambio.

Inicio del visor de ayuda en una MV independiente

Se puede ejecutar el visor 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 visor de ayuda.

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

- 1 Abra el visor de ayuda (Ayuda|Temas de ayuda).
- 2 Seleccione Herramientas|Preferencias 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 visor de ayuda

El Visor de ayuda proporciona muchos métodos rápidos para desplazarse por la documentación de ayuda. Si desea más información, consulte los siguientes temas y la tabla.

- [“Desplazamiento mediante el teclado” en la página 2-14](#)
- [“Cambio entre los paneles del visor de ayuda” en la página 2-14](#)
- [“Adición de pestañas al panel de contenido del Visor de ayuda” en la página 2-14](#)

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

En	Haga esto	Botón	Método abreviado 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 Contenido 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>
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.		

En	Haga esto	Botón	Método abreviado de teclado
Sincronizar la selección de temas en la ficha Contenido con el tema que se ve en el panel de contenido.	Pulse el botón Sincronizar tabla de contenidos.		<i>Ctrl+S</i>
Abrir el tema que precede al tema resaltado en la tabla de contenidos.	Pulse el botón Tema anterior.		<i>Ctrl+Flecha abajo</i>
Abrir el tema que sigue al tema resaltado en la tabla de contenidos.	Pulse el botón Tema siguiente.		<i>Ctrl+Flecha arriba</i>

Desplazamiento mediante el teclado

Cuando se selecciona una entrada de la ficha Contenido, Índice o Buscar del visor de ayuda, las acciones del teclado tienen efecto sobre el lado izquierdo. Las teclas de desplazamiento, como *Re Pág* y *Av Pág*, siguen desplazando la parte izquierda del Visor de ayuda, no el panel de contenido de la derecha.

Para pasar el foco al panel de contenido, situado en el lado derecho del visor de ayuda, pulse la tecla *Tab*. El *Tabulador* pasa del panel seleccionado en el lado izquierdo del visor de ayuda (Contenido, Índice o Buscar) al panel de contenido y después a los botones de desplazamiento. Cuando el foco se encuentra en el panel de contenido se pueden utilizar las teclas *Re Pág* y *Av Pág* para desplazarse.

Cambio entre los paneles del visor de ayuda

El visor de ayuda muestra dos áreas principales: la de la izquierda (panel de navegación o vista en árbol), donde se muestran las fichas Contenido, Índice y buscar, y la de la derecha (panel de contenido). Se puede cambiar de un área a otra mediante un comando del menú contextual del panel de contenido.

Para cambiar entre los paneles del visor de ayuda:

- Haga clic con el botón derecho del ratón en el panel de contenido del visor de ayuda.
 - Elija Ocultar el panel de navegación para ocultar el panel izquierdo.
 - Elija Mostrar el panel de navegación para mostrar los dos paneles.

Adición de pestañas al panel de contenido del Visor de ayuda

Por defecto, el panel de contenido (el de la derecha) del Visor de ayuda muestra un documento cada vez. Es posible añadir varias pestañas al panel de contenido con el fin de ver documentos adicionales y desplazarse a ellos sin cambiar de panel. Cada pestaña muestra un documento distinto.

Para crear pestañas, siga estas instrucciones:

- 1 Seleccione Archivo|Nuevo pestaña.

Después de crear una pestaña también es posible acceder al comando de menú Nueva pestaña eligiendo Nueva pestaña en el menú contextual de las pestañas, o mediante la combinación de teclas *Ctrl+T*.

- 2 Haga clic en el título del documento, en la vista en árbol de la ficha Contenido. (Seleccione el documento que desea presentar en el panel de contenido.)

La nueva ficha muestra el título del documento, y el panel de contenido muestra el documento.

Las pestañas se pueden cerrar con cualquiera de estos métodos:

- Elija Archivo|Cerrar la pestaña o Archivo|Cerrar las otras pestañas.
El comando Cerrar las otras pestañas cierra todas las pestañas del panel de contenido. El comando Cerrar la pestaña sólo cierra la pestaña del documento visible.
- Haga clic con el botón derecho del ratón en una pestaña y elija Cerrar la pestaña o Cerrar las otras pestañas.

Consulta de la documentación de referencia de las clases

La documentación de referencia de las clases se puede consultar en el IDE de JBuilder, en el visor de ayuda o con un navegador web.

Los siguientes métodos permiten presentar la documentación en el IDE:

- 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 abrir el Visor Javadoc, que muestra la documentación de la clase.

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 visor de ayuda, utilice uno de los siguientes métodos:

- En la ventana principal de JBuilder, seleccione Ayuda y la guía deseada.
- En la ficha Contenido del visor de ayuda, haga clic en el enlace de un manual de referencia.

Diseño de la documentación de referencia

La documentación de la *Referencia de la biblioteca de componentes DataExpress* se suministra en formato Javadoc estándar. Si desea información sobre el uso del formato Javadoc, consulte “Organización de este documento API” en `\apiref\help-doc.html`.

Impresión de tutoriales y otros tipos de documentación de JBuilder

Para imprimir los tutoriales y el resto de la documentación de JBuilder, abra el documento deseado en el visor de ayuda y elija ArchivoImprimir. Si desea mostrar una vista preliminar del documento impreso en una ventana independiente, seleccione ArchivoVista preliminar. El Visor de ayuda también permite cambiar el aspecto del documento impreso con el comando Configurar página.

El comando ArchivoConfigurar página muestra el cuadro de diálogo del mismo nombre, donde se pueden configurar los siguientes elementos de diseño:

- Márgenes
- Escala
- Cabeceras/Pies de página
- Orientación vertical
- Orientación apaisada

La documentación también está disponible en formato PDF en la página web de JBuilder, en <http://www.borland.com/techpubs/jbuilder/>.

Si quiere una lista de tutoriales y sus archivos PDF, consulte AyudaTutoriales de JBuilder.

Técnicas adaptativas aceptadas por JBuilder

JBuilder admite técnicas de navegación alternativa por varios motivos: permitir que los usuarios mantengan la forma de trabajo que les resulte más cómoda, con o sin ratón, y proporcionar compatibilidad con estrategias adaptativas, como el software de reconocimiento de voz, que dependen de comandos a los que se puede acceder desde el teclado.

JBuilder acepta las convenciones habituales de desplazamiento mediante el teclado:

- La combinación de teclas *Mayús+F10* abre el menú contextual del objeto que tiene el foco (el lugar del IDE de JBuilder donde se coloque el cursor).
- La tecla *Tab* y la combinación *Ctrl+Tab* para desplazar el foco entre las secciones de la interfaz. Normalmente, esto transmite el foco de izquierda a derecha, en el sentido de las agujas del reloj.
- Las *teclas de cursor* permiten subir, bajar y desplazarse a izquierda y derecha dentro de una sección de la interfaz.
- La *barra de espacio* sirve para activar y desactivar opciones, como un botón circular o una casilla de selección.
- Cuando se pone el foco en un botón y se pulsa la tecla *Intro* se obtiene el mismo efecto que si el botón se hubiera pulsado con el ratón.

- Todos los comandos están disponibles en un menú. Existen teclas aceleradoras para todos los comandos de menú. (Normalmente, se activan pulsando *Alt* + la tecla correspondiente al carácter subrayado en la opción de menú.) Si desea información sobre la forma de mostrar las teclas aceleradoras, consulte la documentación del sistema operativo.

A lo largo de la documentación se describe la forma de utilizar sin ratón las funciones de JBuilder de las que tratan las distintas secciones.

Cumplimiento de la sección 508

La compatibilidad con estas técnicas hace que JBuilder cumpla la sección 508 de la Rehabilitation Act (ley de rehabilitación) de 1973 (código legal estadounidense, 29 794d).

Recursos adicionales para obtener más información sobre JBuilder

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

- El sitio web de JBuilder, que se encuentra en <http://www.borland.com/jbuilder/>, y el de la Comunidad Borland, en <http://community.borland.com/>.
- Si desea una introducción general a JBuilder, abra el Proyecto de bienvenida. Seleccione Archivos\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 las 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.com, 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/>
 <http://www.borland.com/techpubs/jbuilder/>
- Newsletters** Para suscribirse a las newsletters, rellene el formulario en línea que aparece en:
 <http://www.borland.com/products/newsletters/index.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 cree que ha encontrado un error en el software, por favor informe a Borland en alguno de los siguientes sitios:

- La página Support Programs en <http://www.borland.com/devsupport/america/>. Pulse el enlace “Reporting Defects” para llegar al formulario Entry.
- Quality Central en <http://qc.borland.com>. Siga las instrucciones de la sección “Bugs Reports” de la página Quality Central.
- Comando de menú Quality Central del menú Herramientas de JBuilder (Herramientas\Quality Central). Siga las instrucciones para crear una cuenta de usuario de QC e informar del fallo. Si desea más información, consulte la documentación de Borland Quality Central.

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.

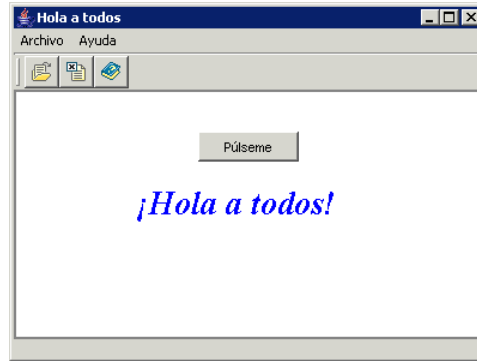
JBuilder es una herramienta creada por desarrolladores y para desarrolladores. Valoramos intensamente sus aportaciones.

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.
- Compilar, crear y ejecutar la aplicación.
- Ejecutar la aplicación desde la línea de comandos.
- Asignar un método de suceso a un botón.
- Completar la interfaz de usuario.
- Agrupar los archivos para la distribución.
- Ejecutar la aplicación distribuida desde el archivo JAR.
- Ejecutar la aplicación distribuida desde la línea de comandos.

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



Si necesita información específica sobre el IDE o el editor mientras trabaja con el tutorial, consulte los siguientes capítulos: [Capítulo 4, “El IDE de JBuilder”](#) y [Capítulo 5, “El editor”](#).

Para ver el código fuente completo, consulte [“Código fuente de HolaATodos” en la página 3-25](#).

El apartado Accesibilidad en las Sugerencias de JBuilder contiene sugerencias sobre la utilización de las funciones de JBuilder para mejorar la facilidad de uso de JBuilder por parte de 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

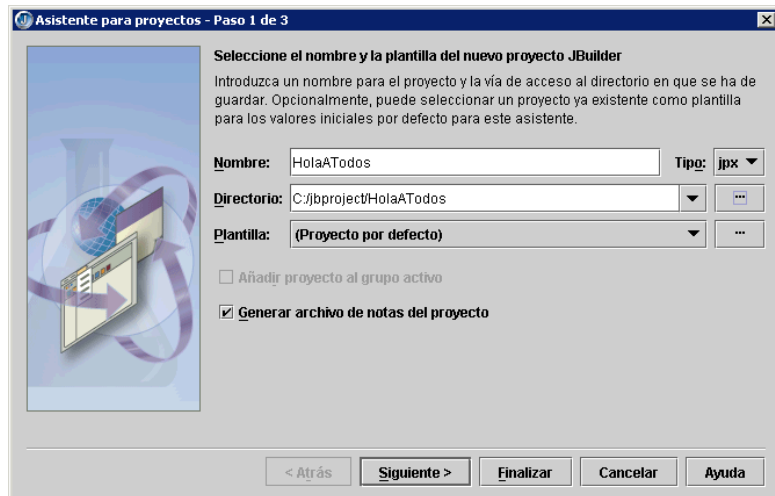
Cuando se escribe el nombre del proyecto en el campo Nombre, se escribe el mismo nombre en el campo Directorio. Por defecto, JBuilder utiliza este nombre de proyecto para crear los nombres del directorio del proyecto y el paquete de las clases. El nombre del paquete se convierte en minúsculas, según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

(Consulte [“Convenciones de la documentación” en la página 2-4.](#)) Para obtener más información sobre los proyectos, consulte el [Capítulo 7, “Proyectos de JBuilder”](#).

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

Quando 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 para proyectos debe tener el siguiente aspecto, con excepción de la vía de acceso de Directorio:



- 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 Paso 2 del asistente ofrece un aspecto semejante a éste:

- 6 Haga clic en **Siguiete** 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.
 - 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:

Etiqueta	Texto
Título:	Hola a todos
Descripción:	Este es el tutorial Hola a todos
Copyright:	Copyright (c) 2004
Empresa:	Mi empresa
@author	Mi nombre
@version	1.0

8 Pulse el botón Finalizar.

En el panel del proyecto situado en la parte superior izquierda del IDE de JBuilder 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 IDE. 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.

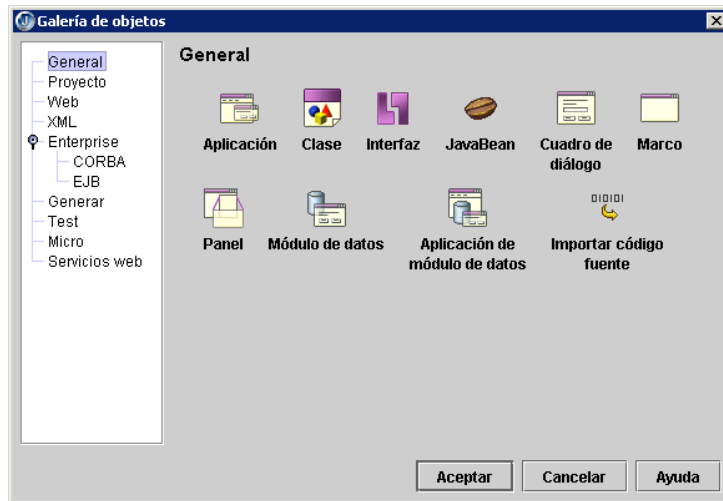
Paso 2: Generación de los archivos de código fuente

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 Para abrir la galería de objetos, elija Archivo|Nuevo o pulse el botón Nuevo de la barra de herramientas principal.



2 Seleccione General y haga doble clic en el icono Aplicación para que se abra el Asistente para aplicaciones.

3 Escriba `HolaATodosClase` en el campo Nombre de clase, en el primer paso del Asistente para aplicaciones.

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

4 Acepte el nombre del paquete por defecto, `holaatodos`.

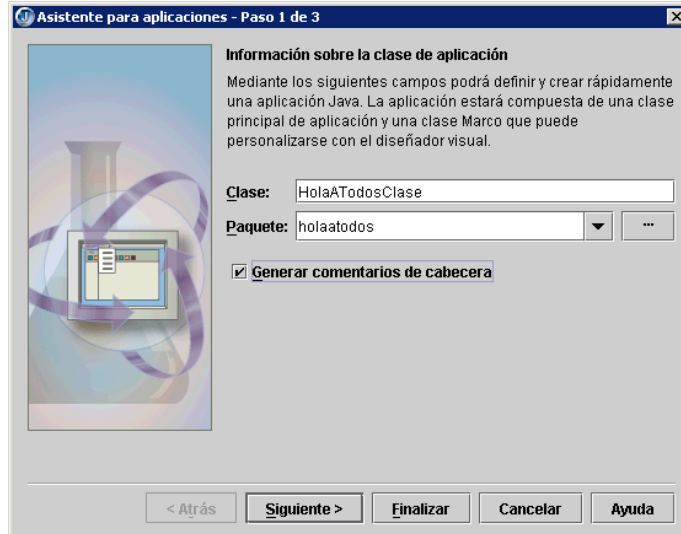
Paso 2: Generación de los archivos de código fuente

Por defecto, el asistente toma el nombre del paquete del nombre del archivo del proyecto, `HolaATodos.jpx`.

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 ofrece el siguiente aspecto:



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

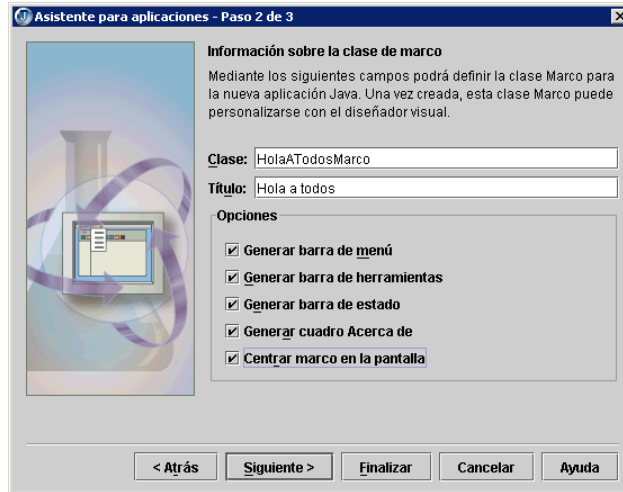
7 Escriba `HolaATodosMarco` en el campo Clase para dar un nombre a la clase de marco.

8 Escriba `Hola a todos` en el campo 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 ofrece 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.

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.

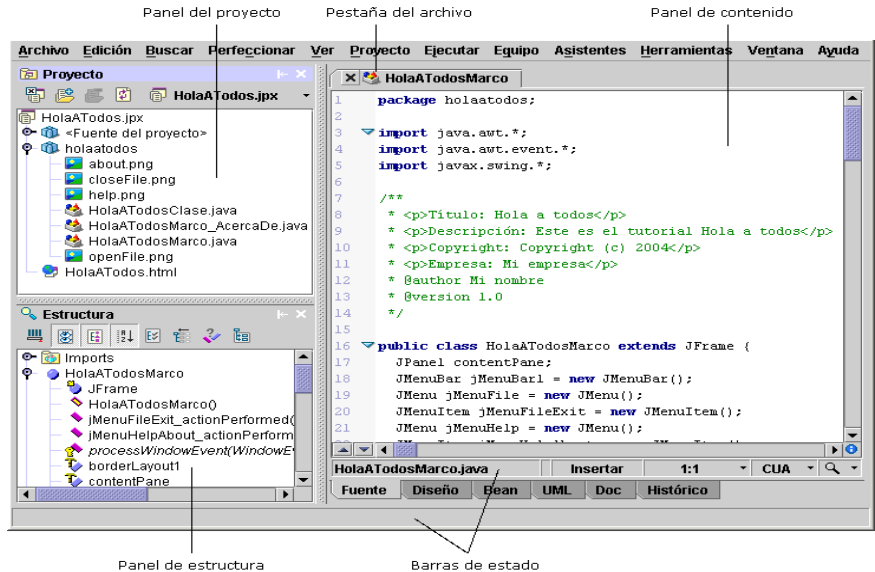
Nota El Paso 3 (la definición y creación de una nueva configuración de ejecución para la aplicación) se omite, ya que no es necesario crear una nueva configuración de aplicación para la aplicación `HolaATodos`. Si desea más información sobre las configuraciones de ejecución, consulte “Definición de las configuraciones de ejecución” en *Creación de aplicaciones con JBuilder*.

Nota En el panel del proyecto 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).

El IDE debe tener un aspecto parecido al de la siguiente captura de pantalla, que muestra el proyecto `Hola a todos`, los archivos, las estructuras y el código fuente.

En la siguiente imagen se muestra el aspecto que debe tener el IDE de JBuilder con el proyecto `Hola a todos`, los archivos, las estructuras y el código fuente en los paneles del proyecto, de estructura y de contenido. (Es posible que vea algo ligeramente distinto, según el sistema operativo y la configuración del espacio de trabajo.)

Figura 3.1 Los elementos del IDE de JBuilder



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

Nota los archivos fuente se guardan en: `<raíz>/jbproject/HolaATodos/src/holaatodos`.

Los archivos de clase generados por el compilador Java a partir de los archivos fuente se guardan en: `<raíz>/jbproject/HolaATodos/classes/holaatodos`

Paso 3: Compilación y ejecución de la aplicación

Ahora, cree y ejecute la aplicación. La generación incluye las tareas de generación, como la preparación de archivos que no son de Java para la compilación, la compilación de los archivos fuente Java, la copia de recursos, la creación de recopilatorios y la distribución, entre otras. La compilación es el proceso de ejecutar el compilador Java. El compilador, que convierte el código fuente en bytecode de Java, genera archivos `.class`.

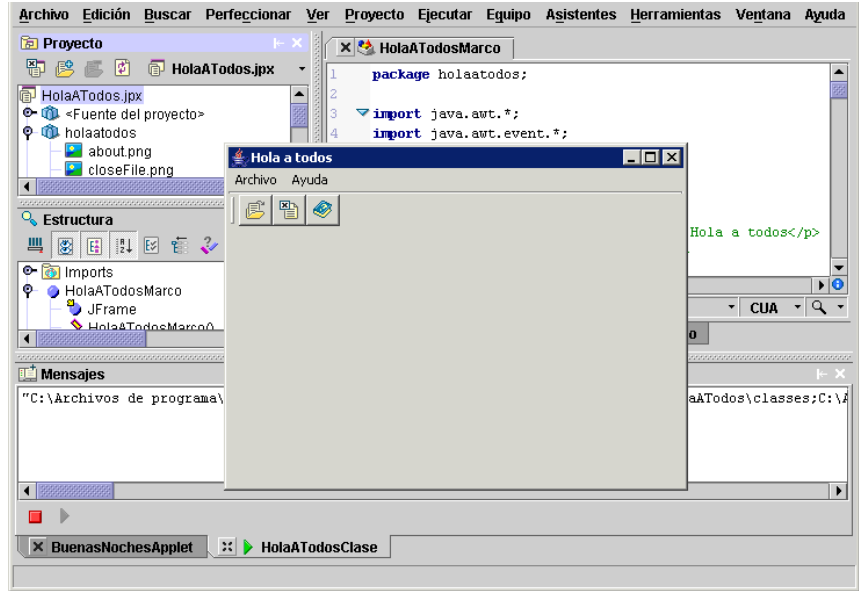


Sugerencia

- 1 Elija Ejecutar|Ejecutar Proyecto, o pulse el botón Ejecutar 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. A continuación aparece la aplicación, que debe ofrecer el siguiente aspecto:



- 2 Seleccione Archivo|Salir en la aplicación “Hola a todos” para cerrarla.
- 3 Pulse el botón Cerrar de la pestaña HolaATodosClase en el panel de mensajes para cerrar los mensajes.

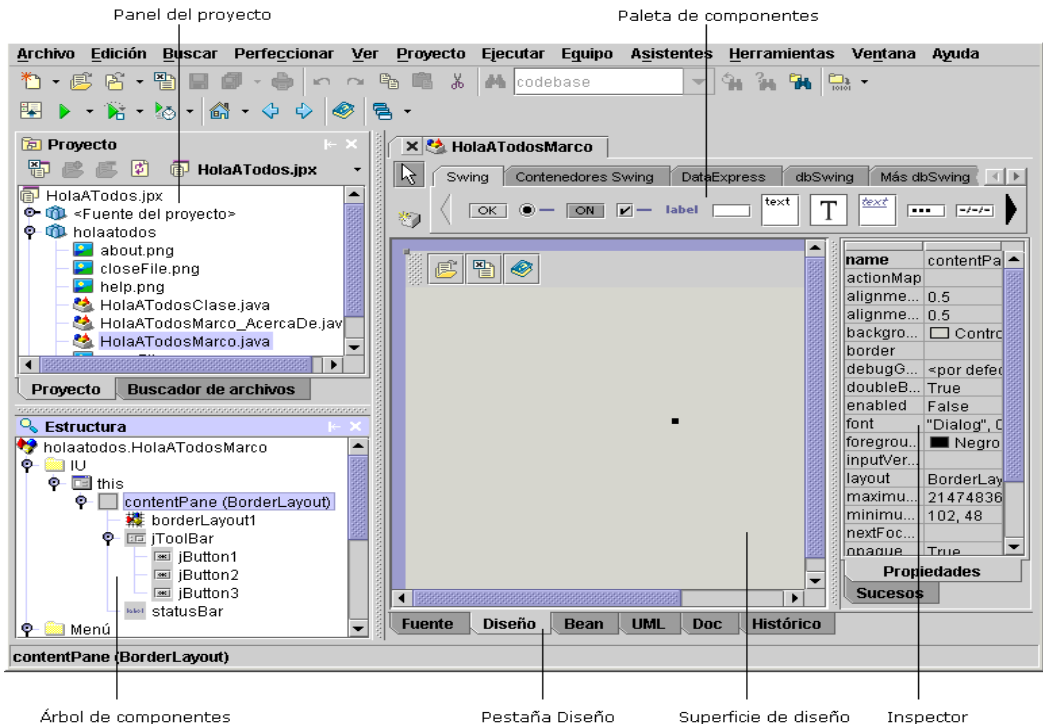
Paso 4: Personalización de la interfaz de usuario del applet

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

- 1 Haga doble clic en `HolaATodosMarco.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 Por defecto.

Figura 3.2 Elementos del diseñador de interfaces 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, su nombre aparece en una etiqueta de ayuda inmediata.

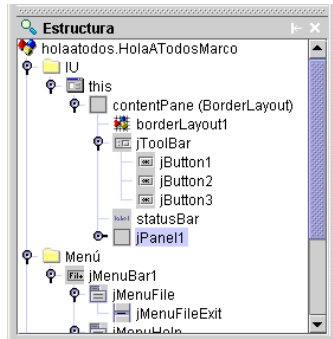


- 4 Haga clic en el centro del marco, en el diseñador de interfaces de usuario, para colocar el componente en el centro del marco.

Nota

La propiedad `constraints` debe tener el valor `Center` en el Inspector. 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 seleccionado en el diseño de la aplicación y en el árbol de componentes.

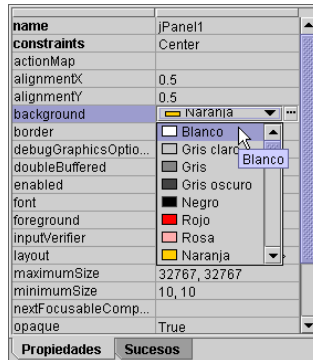


Nota

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

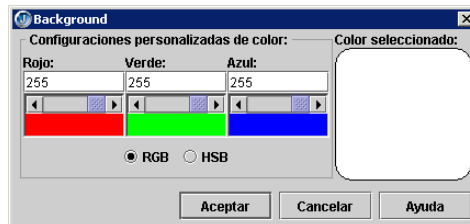
5 Asigne el color blanco al fondo de `jPanel1`.

- 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.

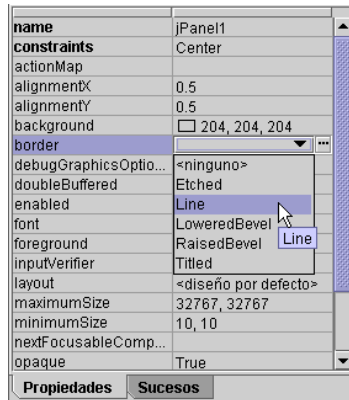


Nota

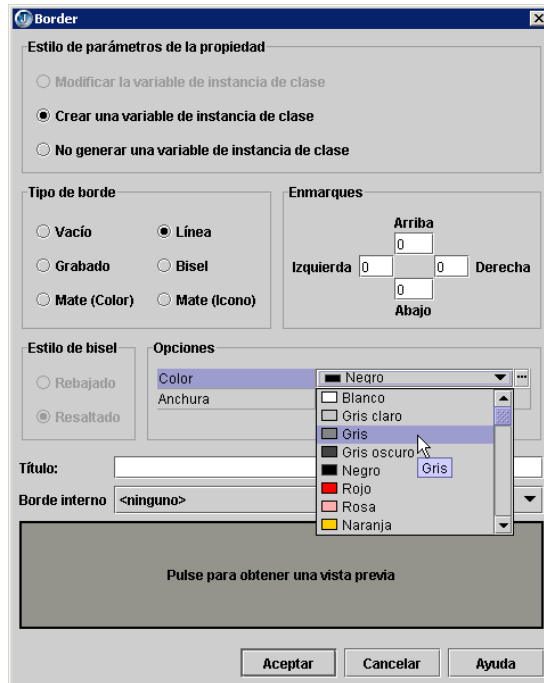
Si hace clic demasiado a la derecha en la columna pulsará el botón de puntos suspensivos, que abre el cuadro de diálogo Fondo. También se puede utilizar para seleccionar el color blanco introduciendo los números 255, que corresponden a los valores de rojo, verde y azul.



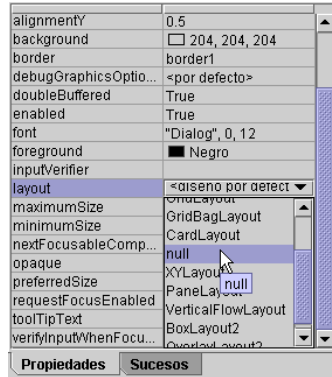
- 6 Añada un borde de línea a jPanel1 y cambie el color del borde a gris.
 - a Haga clic en la columna situada a la derecha de la propiedad `border` en el Inspector.
 - b Haga clic en la *flecha abajo* para abrir la lista desplegable de bordes y seleccione Línea.



- c Pulse el botón de puntos suspensivos para abrir el 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.

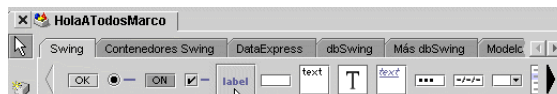


- 8 Seleccione Archivo|Guardar todo o pulse el botón Guardar todo en la barra de herramientas para guardar el proyecto.

Paso 5: Incorporación de componentes a la aplicación

Ahora se va a utilizar la paleta de componentes para añadir una etiqueta al panel.

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

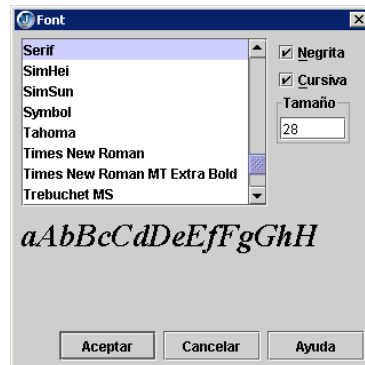


- 2 Coloque el componente en el objeto `JPanel` del diseño, de una de las siguientes maneras:
 - En el árbol de componentes, pulse `JPanel1`. De esta forma se coloca en la esquina superior izquierda del panel.

- 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.

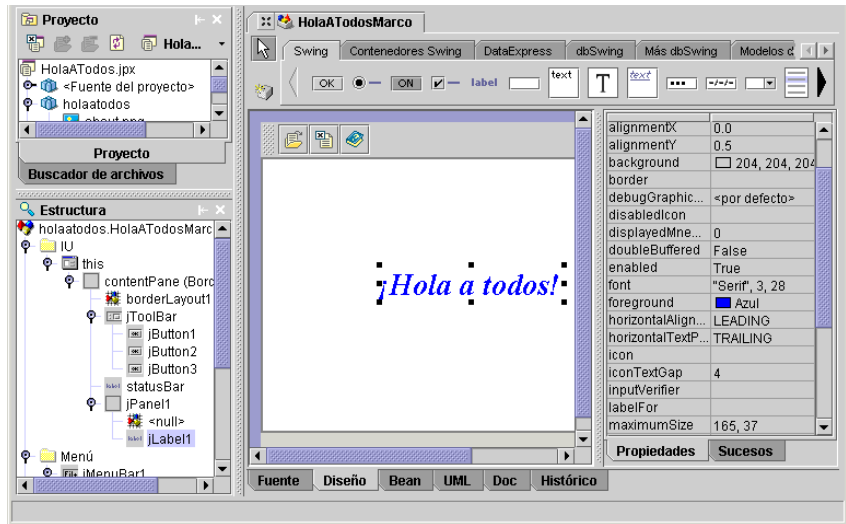
Observe que `JLabel1` se añade debajo de `JPanel1`, en el árbol de componentes. Si coloca el componente en un sitio erróneo, puede seleccionarlo en el árbol de componentes o en el diseñador y pulsar la tecla *Supr*. A continuación, vuelva a intentarlo.

- 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 ¡Hola 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. Intro 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 Pulse 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 botón 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.

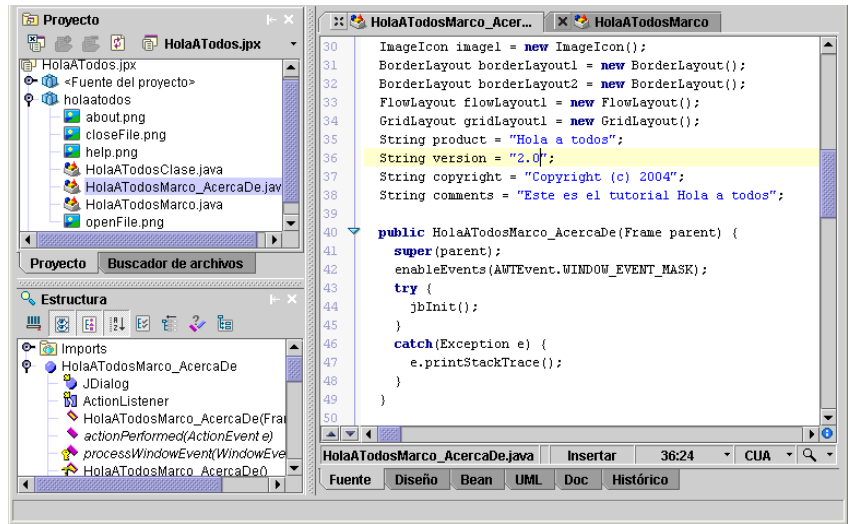
- 2 Seleccione Buscar|Buscar para abrir el cuadro de diálogo Buscar/Reemplazar texto.

- 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 entre las comillas.



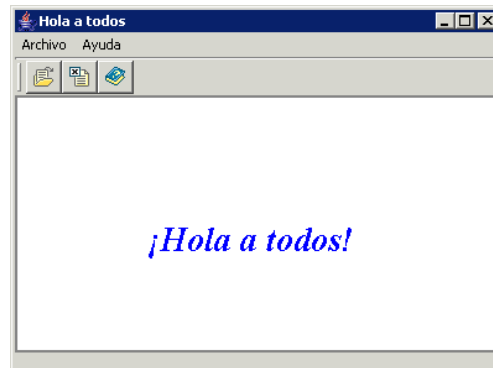
6 Elija Archivo|Guardar todo.

Paso 7: Compilación y ejecución de la aplicación

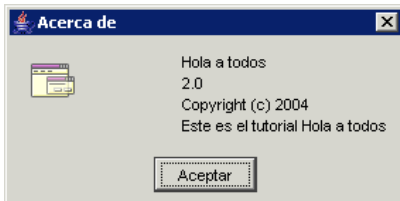
En este punto, la aplicación se puede compilar, crear y ejecutar:

- 1 Seleccione Proyecto|Ejecutar Make del proyecto "HolaATodos.jpx" 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 encontrar en la vía de acceso. Si **java** se encuentra en la vía de acceso, cuando se escribe `java` en la línea de comandos aparece 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:  
/<raíz>/jbproject/HolaATodos/classes holaatodos.HolaATodosClase
```

Nota Es posible que la versión de Windows requiera 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 = /<raíz>/jbproject/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.

- `<raíz>` = 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.



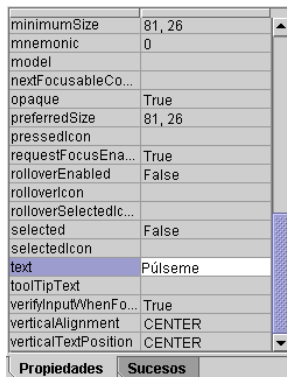
- 2 Haga clic en el componente `JButton`, en la pestaña Swing de la paleta de componentes, y suéltelo en `jPanel1`, en el árbol de componentes o en el panel del diseño. `jButton4` se añade debajo de `jPanel1` en el árbol de componentes.

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



- 4 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 debe presentar el siguiente aspecto:



- 5 Pulse la pestaña Sucesos del Inspector para definir lo que ocurre cuando se pulsa `jButton4`. El objetivo consiste en que el color del texto (propiedad `Foreground`) cambie cuando se pulse el botón.
- 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:

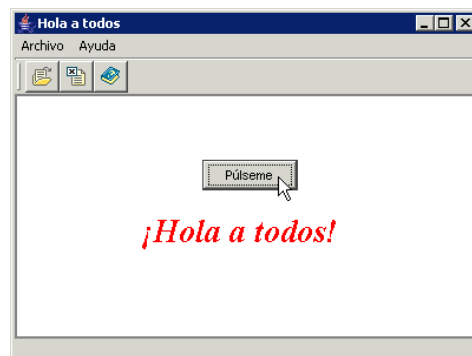
```
void jButton4_actionPerformed(ActionEvent e) {  
    jLabel1.setForeground(new Color(255,0,0));  
}
```

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 Preferencias (Herramientas|Preferencias|Editor). Escriba `setfor` para resaltar `setForeground(Color)` en la ventana emergente, o utilice las teclas de flecha. Pulse *Intro* e introduzca el resto del código que ha aparecido anteriormente.

Ahora, cuando ejecute la aplicación y pulse el botón “Púlseme”, 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 “Púlseme”. El color del texto “¡Hola a todos!” cambia a rojo:



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

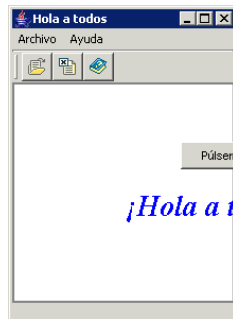
Paso 10: Completar la interfaz de usuario

El último paso para completar la aplicación consiste en aplicarle un diseño portable. Recuerde que si se deja `jPanel1` en el valor `null` o no `layout`, 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 Pulse 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”.

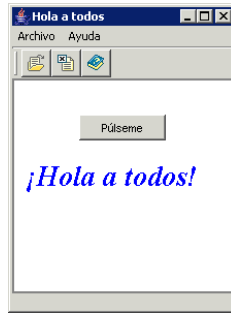
Los diseños portables, a diferencia de `null`, utilizan las posiciones relativas, que son dinámicas. Cuando se emplea un diseño portable, los componentes cambian de posición correctamente cuando el usuario cambia el tamaño de la ventana de la aplicación. En este ejemplo se va a cambiar el diseño a `GridBagLayout`.

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 la propiedad `layout` 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 posición los componentes:



- 7 Cierre la aplicación “Hola a todos”.

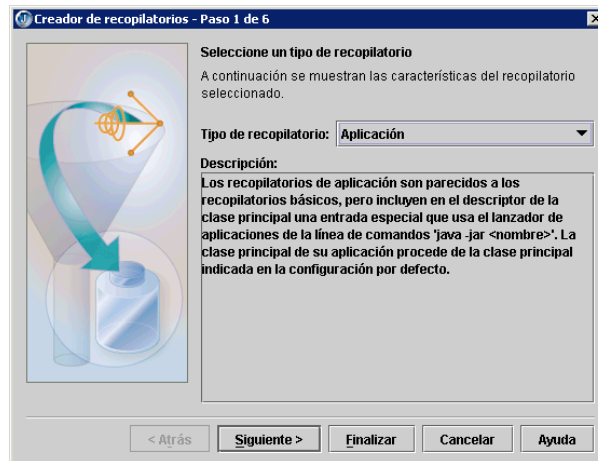
Paso 11: Preparación de la aplicación para la distribución

El Creador de recopilatorios reúne todos los archivos necesarios para distribuir el programa y los recompila 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 en la lista desplegable Tipo de recopilatorio:

El primer paso del Generador de recopilatorios tiene el siguiente aspecto:

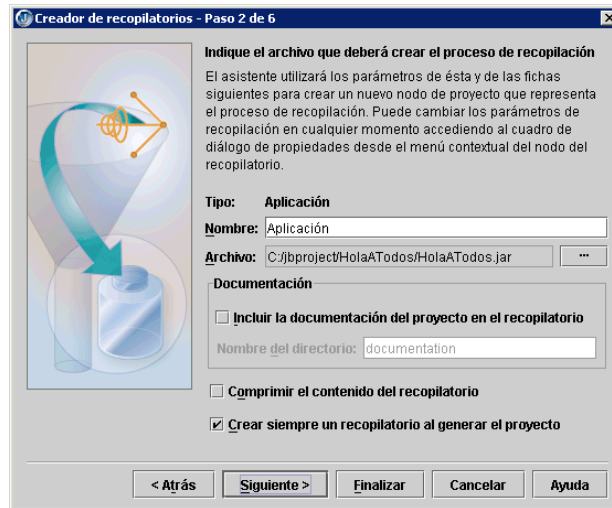


- 3 Pulse Siguiente para pasar al Paso 2.

4 Acepte las opciones por defecto del Paso 2.

Tenga en cuenta que `HolaATodos.jar` se guarda en el directorio `HolaATodos:`

El segundo paso del Creador de recopilatorios tiene el siguiente aspecto:



5 Pulse Finalizar para aceptar los valores por defecto de los siguientes pasos del asistente. Si tiene curiosidad por ver su contenido, vaya pulsando Siguiente en las fichas del Creador de recopilatorios y pulse Finalizar en el último paso. Si desea más información sobre los demás pasos, pulse el botón Ayuda en las fichas.

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

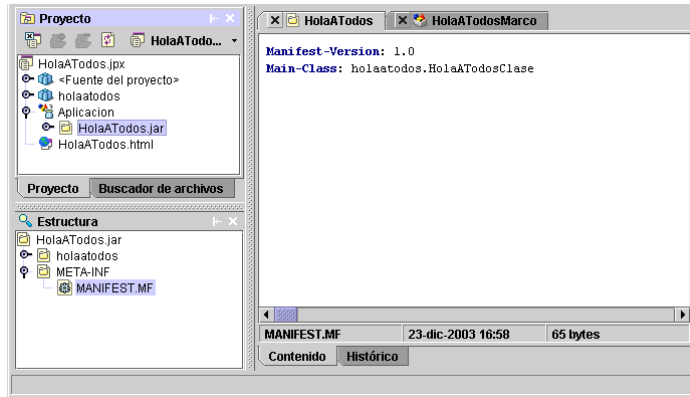
6 Para compilar la aplicación y crear el archivo JAR, haga clic con el botón derecho del ratón el nodo `Aplicación` y elija Ejecutar Make, o seleccione Proyecto|Ejecutar Make del proyecto “`HolaATodos.jpj`”.

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

7 Pulse el icono de ampliación situado junto al nodo del recopilatorio `Aplicación` para ampliarlo y ver el archivo `HolaATodos.jar`.

8 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.



9 Guarde el proyecto.

Si desea más información sobre la distribución, consulte “Distribución de programas en Java”, en *Creación de aplicaciones con JBuilder*.

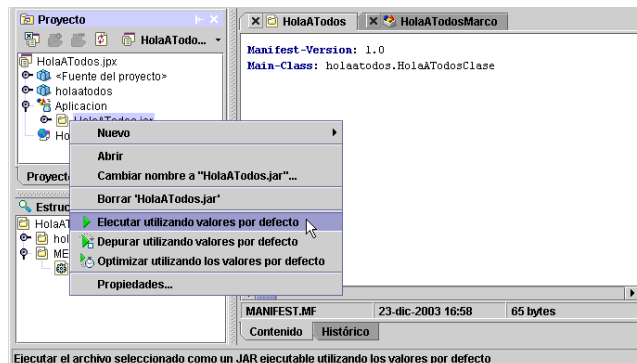
Paso 12: Ejecución de la aplicación distribuida desde el archivo JAR

**Es una función de
JBuilder Developer y
Enterprise.**

La aplicación distribuida se puede comprobar mediante un proceso muy fácil: ejecútela con el archivo JAR ejecutable creado en el paso anterior con el Creador de compiladores. El panel del proyecto muestra el archivo JAR, `HolaATodos.jar`, dentro del nodo `Aplicación`.

- 1 Amplíe el nodo `Application` en el panel del proyecto.
- 2 Haga clic con el botón derecho del ratón en el archivo `HolaATodos.jar`.
- 3 Elija `Ejecutar` utilizando valores por defecto.

La aplicación `Hola a todos` se carga y se ejecuta (aparece el cuadro de diálogo `Hola a todos`, con el botón `Púlsame` en el panel central).



Paso 13: 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 encontrar en la vía de acceso. Si **java** se encuentra en la vía de acceso, cuando se escribe `java` en la línea de comandos aparece 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 -jar /<home>/jbproject/HolaATodos/HolaATodos.jar
```

Nota en la plataforma Windows se utiliza la barra invertida (\).

El comando debe tener la siguiente estructura:

```
java -jar classpath
```

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

En el ejemplo:

- `java` = ejecuta la aplicación en Java.
- `jar` = una opción que establece la vía de acceso de búsqueda del archivo JAR (que contiene las clases de la aplicación, incluida la principal, y los recursos).
- `<raíz>` = el directorio inicial, como por ejemplo, `c:\winnt\profiles\<nombre de usuario>` o `c:\Documents and Settings\<nombre de usuario>`
- `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 compiladores lo guarda en este directorio.

Si desea más información sobre el Creador de compiladores, consulte “El Creador de compiladores”, en *Creación de aplicaciones con JBuilder*.

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>.

¡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

Para ver el código fuente, consulte:

- HolaATodosMarco.java en la [página 3-25](#)
- HolaATodosClase.java en la [página 3-27](#)
- HolaATodosMarco_AcercaDe.java en la [página 3-28](#)

Código fuente de HolaATodosMarco.java

```
package holaatodos;

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

/**
 * Título:      Hola a todos
 * Descripción: Éste es el tutorial "Hola a todos".
 * Copyright:   Copyright (c) 2003
 * Empresa:    Nombre de la empresa
 * @autor Nombre del usuario
 * @versión 1.0
 */

public class HolaATodosMarco 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();
    JLabel jLabel1 = new JLabel();
    JButton jButton4 = new JButton();
    GridBagLayout gridBagLayout1 = new GridBagLayout();

    //Construir el marco
    public HolaATodosMarco() {
        enableEvents(AWTEvent.WINDOW_EVENT_MASK);
        try {
            jbInit();
        }
        catch(Exception e) {
            e.printStackTrace();
        }
    }
    //Inicialización del componente
    private void jbInit() throws Exception {
        image1 = new ImageIcon(holaatodos.HolaATodosMarco.class.getResource("openFile.png"));
        image2 = new
            image2 = new
        ImageIcon(holaatodos.HolaATodosMarco.class.getResource("closeFile.png"));
    }
}
```

```

image3 = new ImageIcon(holaatodos.HolaATodosMarco.class.getResource("help.png"));
contentPane = (JPanel) this.getContentPane();
contentPane.setLayout(borderLayout1);
this.setSize(new Dimension(400, 300));
this.setTitle("Hola a todos");
statusBar.setText(" ");
jMenuFile.setText("Archivo");
jMenuFileExit.setText("Salir");
jMenuFileExit.addActionListener(new
HolaATodosMarco_jMenuFileExit_ActionAdapter(this));
jMenuHelp.setText("Ayuda");
jMenuHelpAbout.setText("Acerca de");
jMenuHelpAbout.addActionListener(new
HolaATodosMarco_jMenuFileExit_ActionAdapter(this));
jButton1.setIcon(image1);
jButton1.setToolTipText("Abrir archivo");
jButton2.setIcon(image2);
jButton2.setToolTipText("Cerrar archivo");
jButton3.setIcon(image3);
jButton3.setToolTipText("Ayuda");
jLabel1.setFont(new java.awt.Font("Serif", 3, 28));
jLabel1.setForeground(Color.blue);
jLabel1.setText(";Hola a todos!");
jPanel1.setLayout(gridBagLayout1);
jPanel1.setBackground(Color.white);
jButton4.setPreferredSize(new Dimension(81, 26));
jButton4.setText("Púlsame");
jButton4.addActionListener(new HelloWorldFrame_jButton4_actionAdapter(this));
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(statusBar, BorderLayout.SOUTH);
contentPane.add(jToolBar, BorderLayout.NORTH);
contentPane.add(jPanel1, BorderLayout.CENTER);
jPanel1.add(jButton4, new GridBagConstraints(0, 0, 1, 1, 0.0, 0.0,
GridBagConstraints.CENTER, GridBagConstraints.NONE, new Insets(49, 156,
0, 159), 4, 0));
jPanel1.add(jLabel1, new GridBagConstraints(0, 1, 1, 1, 0.0, 0.0,
GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(19, 126, 133,
113), 10, -12));
}
//Realizar Archivo | Salir
public void jMenuFileExit_actionPerformed(ActionEvent e) {
System.exit(0);
}
//Realizar Ayuda | Acerca de
public void jMenuHelpAbout_actionPerformed(ActionEvent e) {
HolaATodosMarco_AcercaDe dlg = new HolaATodosMarco_AcercaDe(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.pack();
dlg.show();
}
//Modificado para poder salir cuando se cierra 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));
    }
}

class HolaATodosFrame_jMenuFileExit_ActionAdapter implements ActionListener {
HolaATodosFrame adaptee;

HolaATodosFrame_jMenuFileExit_ActionAdapter(HolaATodosFrame adaptee) {
this.adaptee = adaptee;
}

public void actionPerformed(ActionEvent e) {
adaptee.jMenuFileExit_actionPerformed(e);
}
}

class HolaATodosFrame_jMenuHelpAbout_ActionAdapter implements ActionListener {
HolaATodosFrame adaptee;

HolaATodosFrame_jMenuHelpAbout_ActionAdapter(HolaATodosFrame adaptee) {
this.adaptee = adaptee;
}

public void actionPerformed(ActionEvent e) {
adaptee.jMenuHelpAbout_actionPerformed(e);
}
}

class HolaATodosFrame_jButton4_actionAdapter implements java.awt.event.ActionListener {
HolaATodosFrame adaptee;

HolaATodosFrame_jButton4_actionAdapter(HolaATodosFrame adaptee) {
this.adaptee = adaptee;
}

public void actionPerformed(ActionEvent e) {
adaptee.jButton4_actionPerformed(e);
}
}

```

Código fuente de HolaATodosClase.java

```

package holaatodos;

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

/**
 * Título:      Hola a todos
 * Descripción: Éste es el tutorial "Hola a todos".
 * Copyright:   Copyright (c) 2003
 * Empresa:     Nombre de la empresa
 * @autor Nombre del usuario
 * @versión 1.0
 */

public class HolaATodosClase {
    boolean packFrame = false;

    // Crear la aplicación
    public HolaATodosClase() {
        HolaATodosMarco frame = new HolaATodosMarco();
        //Valida los marcos que tienen tamaños preestablecidos
        //Empaquetar los marcos con información de tamaño preferente de los usuarios, 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 principal
public static void main(String[] args) {
    try {
        UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName());
    }
    catch(Exception e) {
        e.printStackTrace();
    }
    new HolaATodosMarco();
}
```

Código fuente de HolaATodosMarco_AcercaDe.java

```
package holaatodos;

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

/**
 * Título:      Hola a todos
 * Descripción: Éste es el tutorial "Hola a todos".
 * Copyright:   Copyright (c) 2003
 * Empresa:     Nombre de la empresa
 * @autor Nombre del usuario
 * @versión 1.0
 */

public class HolaATodosMarco_AcercaDe 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();
    ImageIcon imagel = new ImageIcon();
    BorderLayout borderLayout1 = new BorderLayout();
    BorderLayout borderLayout2 = new BorderLayout();
    FlowLayout flowLayout1 = new FlowLayout();
    GridLayout gridLayout1 = new GridLayout();
    String product = "Hola a todos";
    String version = "2.0";
```

```

String copyright = "Copyright (c) 2003";
String comments = "Éste es el tutorial de Hola a todos";
public HolaATodosMarco_AcercaDe(Frame parent) {
    super(parent);
    enableEvents(AWTEvent.WINDOW_EVENT_MASK);
    try {
        jbInit();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}

//Inicialización del componente
private void jbInit() throws Exception {
    image1 = new ImageIcon(holaatodos.HolaATodosMarco.class.getResource("about.png"));
    imageLabel.setIcon(image1);
    this.setTitle("Acerca de");
    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("Ok");
    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);
    setResizable(true);
}

//Modificado para poder salir cuando se cierra la ventana
protected void processWindowEvent(WindowEvent e) {
    if (e.getID() == WindowEvent.WINDOW_CLOSING) {
        cancel();
    }
}

super.processWindowEvent(e);
}

//Cerrar el cuadro de diálogo
void cancel() {
    dispose();
}

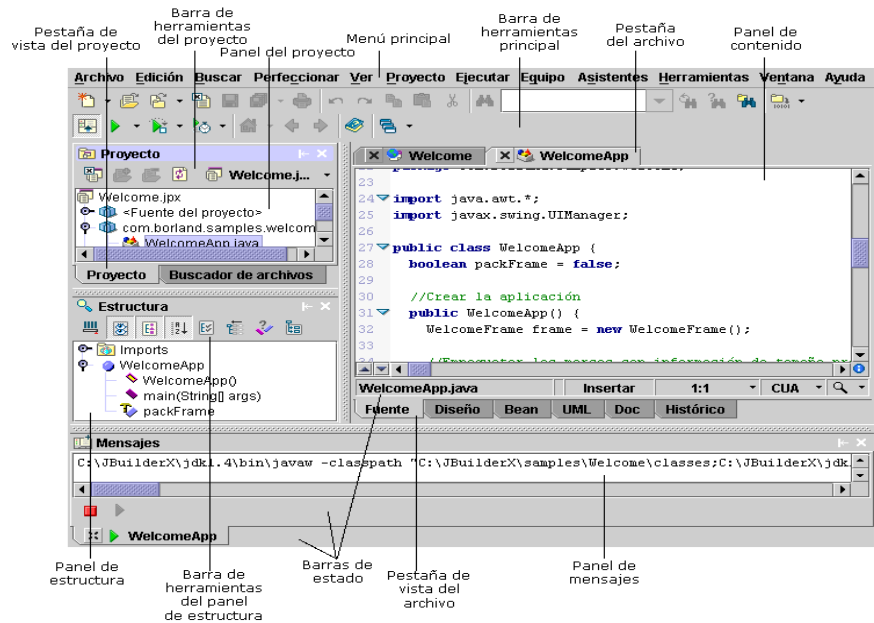
//Cerrar el cuadro de diálogo cuando se ejecuta un suceso de botón
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == button1) {
        cancel();
    }
}
}

```


El IDE de JBuilder

JBuilder sólo utiliza una ventana para realizar la mayoría de las funciones de desarrollo: modificación, diseño visual, desplazamiento, búsqueda, compilación, depuración, etc. Esta ventana es el IDE de JBuilder (también se denomina “visualizador de JBuilder”) y contiene varios paneles en los que se efectúan estas funciones de desarrollo. Los paneles que constituyen la ventana de JBuilder se pueden segregar. Si desea más información sobre los paneles móviles y los espacios de trabajo configurables de JBuilder, consulte [“Configuración del espacio de trabajo” en la página 4-6](#). 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 del IDE



Estos son los componentes del IDE de JBuilder:

- Menú principal
- Barra de herramientas principal
- Configuraciones del espacio de trabajo
- Panel del proyecto
- Panel de estructura
- Panel de contenido
- Panel de mensajes
- Depurador
- Barras de estado

Es posible abrir varias instancias de JBuilder (es decir, de la ventana o el IDE de JBuilder) y arrastrar una copia de un archivo a otro panel del proyecto, en otra instancia de JBuilder. Todas las instancias del Visualizador de aplicaciones muestran los mismos proyectos abiertos, pero se pueden abrir archivos distintos en los paneles de contenido de cada proyecto. Utilice el menú Ver para decidir qué paneles desea que se muestren en el IDE actual.

El IDE proporciona opciones de desplazamiento, búsqueda, definición de favoritos y personalización. Para obtener más información consulte los siguientes temas:

- [“Desplazamiento y búsqueda en el IDE” en la página 4-27](#)
- [“Búsqueda en árboles” en la página 4-28](#)
- [“Búsqueda de clases” en la página 4-29](#)
- [“Utilización de Favoritos” en la página 4-30](#)

- [“Personalización del IDE” en la página 4-32](#)

JBUILDER proporciona numerosas formas de ajustar el IDE. Entre otras, destacan las siguientes:

- Determinación de los conjuntos de funciones que se muestran (consulte “Página de personalidad (cuadro de diálogo Propiedades del proyecto) en la ayuda en línea).
- Configuración de la disposición de los paneles que constituyen el IDE (consulte [“Configuración del espacio de trabajo” en la página 4-6](#)).
- Configuración de las vías de acceso, las opciones de generación y el formato que utiliza JBUILDER (consulte “Definición de las propiedades del proyecto” en *Creación de aplicaciones con JBUILDER*).
- Personalización del comportamiento y la apariencia del editor (consulte “Personalización del editor” en la ayuda en línea).

Consulte

- El botón Ayuda de la ficha Personalidad del cuadro de diálogo Propiedades de proyecto (Proyecto|Propiedades de proyecto|Personalidad)
- [“Configuración del espacio de trabajo” en la página 4-6](#)
- “Definición de las propiedades del proyecto” en *Creación de aplicaciones con JBUILDER*
- “Personalización del IDE de JBUILDER” en la ayuda en línea

Menú principal

El menú principal se encuentra en la parte superior del IDE de JBUILDER. Si desea información sobre los distintos menús, consulte “Menús de JBUILDER”.

Archivo Edición Buscar Perfeccionar Ver Proyecto Ejecutar Equipo Asistentes Herramientas Ventana

Para acceder al índice de la ayuda en línea:

- Seleccione Ayuda|Temas de ayuda.
- Haga clic en la pestaña Índice.
- Escriba `principal` en el campo Escriba las primeras letras del entrada.

Barra de herramientas principal

La barra de herramientas principal se encuentra en la parte superior del IDE, 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. Si mueve el cursor sobre el botón, puede ver también el nombre y una breve descripción del botón de la barra de herramientas. Debajo del botón aparece su nombre, y una breve descripción aparece en la barra de estado principal.

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; 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 Iconos de la barra de herramientas principal








Icono	Equivalente en el menú	Descripción
	Archivo Nuevo	Abre la galería de objetos en la que se puede elegir entre distintos asistentes. La flecha contigua a Nuevo abre un menú desplegable con la opción Nuevo (abre la galería de objetos) y acceso a los 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. La flecha del icono Guardar todo abre un menú desplegable que incluye el comando Guardar todo y la opción de guardar el archivo abierto actualmente.
	Archivo Imprimir	Imprime el archivo o el texto seleccionado.

Tabla 4.1 Iconos de la barra de herramientas principal (continuación)




















Icono	Equivalente en el menú	Descripción
	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 el efecto de Deshacer. La acción Rehacer cuenta con muchos niveles.
	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 la clase especificada en el visualizador de JBuilder. La clase debe encontrarse en la vía de acceso de importación del archivo actual.
	Proyecto Ejecutar	Compila los recursos y los archivos <code>.java</code> del proyecto seleccionado, si los archivos <code>.class</code> correspondientes no existen o han quedado obsoletos. También compila los archivos importados de los que depende el proyecto, si sus archivos <code>.class</code> 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.
	Ejecutar Ejecutar el 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.

Tabla 4.1 Iconos de la barra de herramientas principal (continuación)

Icono	Equivalente en el menú	Descripción
	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 instalado.
	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.
	Ver Selector de espacios de trabajo	Guarda, restablece, gestiona y configura el espacio de trabajo en la configuración por defecto.

Configuración del espacio de trabajo

La configuración del espacio de trabajo consiste en disponer los paneles y las pestañas del IDE de la forma más cómoda para el usuario. Es posible guardar distintas configuraciones de trabajo que resulten útiles para diversos tipos de tareas.

JBuilder acepta la maximización y el restablecimiento de paneles; la iconización, la integración y la segregación de paneles; la división del panel de contenido entre varios archivos y la división de las pestañas de archivo para presentar varios archivos simultáneamente.

Los paneles se desplazan arrastrando su barra de título. Las barras de título de los paneles

- Aceptan el método de arrastrar y colocar y el doble clic.
- Contienen iconos que proporcionan acciones de desplazamiento específicas.
- Proporcionan un menú contextual (que se abre haciendo clic con el botón derecho) para las acciones de desplazamiento (también está disponible en Ver|Acciones de ventana).

Estos comandos son sensibles al contexto, y varían según el estado y la posición actuales del panel.

Los comandos que controlan las configuraciones del espacio de trabajo están disponibles en dos lugares:

- Seleccione **Ver**|Espacios de trabajo y seleccione un comando del submenú.
- Pulse la flecha abajo contigua al icono Espacios de trabajo de la barra de herramientas y seleccione un comando del menú desplegable.



Los dos menús contienen los mismos comandos. Dado que los dos menús permiten hacer exactamente lo mismo, normalmente se hace referencia a cualquiera de ellos como *menú Espacios de trabajo*.

El elemento de menú “por defecto” del menú Espacios de trabajo es la configuración del espacio de trabajo que se ve la primera vez que se inicia JBuilder. Si desea volver a la disposición original de JBuilder en cualquier momento, seleccione **Por defecto** en el menú Espacios de trabajo y active un archivo. Si desea realizar cambios en una configuración que le resulta útil, asígnele antes un nombre. De esta forma estará disponible en el menú Espacios de trabajo y será posible pasar a ella en cualquier momento.

Cuando se cierra JBuilder sin guardar la configuración actual del espacio de trabajo, JBuilder la guarda automáticamente como **Workspace1**. La nueva configuración se activa cuando se vuelve a iniciar JBuilder. La siguiente vez que se cierra JBuilder sin guardar una configuración en la que se han realizado cambios, ésta se convierte en **Workspace1**.

Nota

Las configuraciones del espacio de trabajo no son lo mismo que las configuraciones de personalidad, una función de JBuilder Enterprise. Las configuraciones de personalidad determinan las tecnologías visibles en el IDE. Si desea más información sobre las configuraciones de personalidad, elija **Proyecto**|**Propiedades** de **proyecto**|**Personalidad** y pulse el botón **Ayuda**.

La configuración “clásica” de JBuilder

Si no desea que los paneles se desplacen ni perder espacio a causa de las barras de título, utilice la configuración “clásica” de JBuilder:

- 1 Seleccione **Herramientas**|**Preferencias**|**Visualizador**.
- 2 Haga clic en el campo **Diseño del IDE**.
- 3 Seleccione **Paneles fijos**.
- 4 Pulse **Aceptar**.

Los paneles recuperan sus posiciones originales y las barras de título desaparecen de los paneles. Si desea volver a los espacios de trabajo configurables, elija **Paneles personalizables**.

Gestión de las configuraciones

Es posible cambiar de nombre, borrar y devolver a un estado anterior las configuraciones del espacio de trabajo, así como cambiar entre ellas.

Para abrir una configuración con nombre, selecciónela directamente en el menú Espacios de trabajo o elija Seleccionar espacio de trabajo en el menú Espacios de trabajo y seleccione una configuración de la lista.

Elija Restaurar espacio de trabajo en el menú Espacios de trabajo para integrar el panel del proyecto y el buscador de archivos y para ocultar los paneles de estructura y mensajes hasta que los abra el archivo activo. Restaurar espacio de trabajo no modifica la configuración de los archivos divididos.

Modificación de las configuraciones

Cuando se sienta satisfecho con una configuración:

1 Elija Guardar espacio de trabajo como en el menú Espacios de trabajo.

Aparece el cuadro de diálogo Guardar espacio de trabajo como.

2 Después:

- Para crear una configuración con nombre, escriba un nombre nuevo.
- Para sobrescribir una configuración, seleccione su nombre en la lista desplegable.

3 Pulse Aceptar.

El nombre de la nueva configuración se muestra en el menú Espacios de trabajo.

Seleccione Espacios de trabajo|Gestionar espacios de trabajo para abrir el cuadro de diálogo Gestionar espacios de trabajo. Aquí se pueden cambiar de nombre y borrar las configuraciones, así como determinar el orden por el que aparecen en el menú Espacios de trabajo:

- Si desea cambiar el nombre de una configuración, selecciónela y pulse Cambiar nombre.
- Si desea eliminar una configuración, selecciónela y pulse Eliminar.
- Si desea desplazar una configuración a otro lugar, selecciónela y pulse Desplazar hacia arriba o Desplazar hacia abajo.

Cuando esté satisfecho con la lista de configuraciones del espacio de trabajo, pulse Aceptar para cerrar el cuadro de diálogo y aplicar los cambios.

Maximización y restablecimiento de los paneles

Cuando se maximiza un panel, éste ocupa todo el espacio de la ventana del IDE. Cuando se restablece, recupera el tamaño y la posición que tenía antes de que se maximizara. Esto resulta especialmente útil, por ejemplo, en la comprobación

de la salida del panel de mensajes, cuando se desea ver el volcado de una pila u otra salida y volver rápidamente al editor.

Existen varias formas de acceder a los comandos Maximizar y Restablecer.

- Haga doble clic en la barra de título del panel para maximizarlo. Vuelva a hacer doble clic en la barra para restablecer el panel.
- Haga clic en el icono que identifica el panel (a la izquierda de la barra de título) y elija Maximizar o Restablecer en el menú contextual.
- Elija VerAcciones de ventanalMaximizar o Restablecer.
- Pulse las teclas *Alt+F10*.

Iconización, integración y segregación de paneles

Estos comandos permiten controlar la ubicación y la visibilidad de los paneles de proyecto, estructura y mensajes:

- Cuando se iconiza un panel, éste se oculta pero sigue estando disponible en el icono del margen izquierdo de la ventana del IDE. Cuando se pulsa el icono de un panel, éste se abre segregado, esto es, como ventana flotante, dentro de la ventana principal del IDE. Si desea iconizar un panel, pulse el icono Iconizar.
- Cuando se segrega un panel, éste se separa de la ventana principal del IDE y se puede desplazar a cualquier parte de la pantalla, fuera del resto del IDE. Para segregar un panel, haga clic en él y arrástrelo.
- Cuando se integra un panel, éste vuelve a la ventana principal del IDE. Para integrar un panel, pulse el icono Integrar o arrastre el panel a un lugar de anclaje y suéltelo.

También es posible acceder a los comandos que permiten manipular el panel que tiene el foco haciendo clic con el botón derecho del ratón en su barra de título o eligiendo VerAcciones de ventana (*Alt+F10* en el teclado).

Iconización

Si desea abrir y segregar un panel iconizado, pulse una vez el icono. Vuelva a pulsar el icono, sin mover el panel, para volver a iconizarlo. Si lo prefiere, pulse *Alt+1* para controlar el primer panel iconizado, *Alt+2* para controlar el segundo y así sucesivamente.

Integración

A la hora de iconizar paneles, piense que la ventana principal del IDE tiene cuatro sentidos cardinales y dos ordinales: norte, sur, este y oeste, y en el borde izquierdo, noroeste y sudoeste. La posición predeterminada del panel del proyecto es el noroeste, la del panel de estructura es el sudoeste y la del panel de mensajes es el sur.

Los paneles se puede apilar en vertical, en horizontal o de las dos formas, según el borde integrado. Para controlar el lugar en el que se integra el panel, pulse la tecla *Ctrl* y arrastre el panel al lugar donde desea colocarlo. Se

muestran unos indicadores de referencia de color gris oscuro que muestran dónde se integrará el panel en cada momento. Suelte el botón del ratón cuando los indicadores de referencia ocupen la posición deseada.

Presentación de varios archivos

Es posible ver varios archivos a la vez con una sola instancia de JBuilder. Sólo hay un panel de contenido por proyecto, pero es posible abrir varios visores de archivos a la vez. Se puede dividir el panel de contenido entre varios archivos, así como desplazar un archivo a su propio *panel de presentación*. Los paneles de presentación son nuevas ventanas en las que se muestran los archivos segregados. Contienen el menú principal y la barra de herramientas de JBuilder y todas las pestañas normales de presentación de archivos.

Las pestañas de los archivos se pueden arrastrar para dividir el panel de contenido entre varios archivos, así como para segregar un archivo en su propio panel de presentación.

- Cuando el puntero se encuentra en una ubicación que divide el panel de contenido aparecen indicadores de referencia de color gris oscuro.
- Los indicadores de referencia desaparecen cuando el cursor se encuentra en una ubicación que segrega el archivo.

También es posible segregar archivos desde el menú contextual. Para ello, haga clic con el botón derecho del ratón en su pestaña y seleccione Segregar.

El panel de presentación es una ventana independiente, por lo que tiene en la barra de título los mismos controles que cualquier ventana convencional del sistema operativo. Los comandos específicos del espacio de trabajo sólo se encuentran disponibles en el menú contextual (clic derecho) de la pestaña del archivo.

Cuando un archivo se segrega en un panel de presentación, otros archivos pueden compartirlo como pestañas, pero no es posible presentar varios a la vez. Los paneles de presentación segregados no se pueden dividir.

Los archivos se pueden volver a integrar de las siguientes formas:

- Arrastre la pestaña del archivo dentro del visor, al panel de contenido principal. Suelte cuando los indicadores de referencia de color gris oscuro aparezcan alrededor del panel de contenido.
- Haga clic con el botón derecho del ratón en la pestaña del archivo y seleccione Integrar.

Cuando se arrastra la barra de título del visor se desplaza el visor completo, pero el archivo no se vuelve a integrar.

Cuando se arrastra la pestaña de un archivo fuera de la ventana principal del IDE, se dice que se *divide* la pestaña.

Nota Las configuraciones de los paneles de presentación (la división del panel de contenido y la división de las pestañas de archivos) no se guardan dentro de

las configuraciones del espacio de trabajo, sino como parte del proyecto. La configuración de los paneles de presentación no varía cuando se cambia de configuración del espacio de trabajo.

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 proporciona acceso a dos vistas por defecto con estas pestañas de presentación: las pestañas de vista Proyecto y de Buscador de archivos. Sin embargo, si se segrega el panel de estructura o el panel de mensajes y se desplaza al panel del proyecto, éste incluye una pestaña de vista Estructura que muestra la barra de título del panel de estructura y una barra de herramientas, o el panel de mensajes con su barra de título. Si desea obtener más información, consulte [“Panel de estructura” en la página 4-17](#) o [“Panel de mensajes” en la página 4-24](#). Si desea más información sobre los proyectos y el panel del proyecto, elija uno de los siguientes temas.

- [“La pestaña de vista Proyecto del panel del proyecto” en la página 4-11](#)
- [“La pestaña de vista Buscador de archivos del panel del proyecto” en la página 4-12](#)
- [“Ejecución de aplicaciones desde el panel del proyecto” en la página 4-12](#)
- [“Creación de proyectos” en la página 4-13](#)
- [“Adición de archivos, clases, paquetes o aplicaciones a un proyecto” en la página 4-13](#)
- [“El panel del proyecto” en la página 4-16](#)

La pestaña de vista Proyecto del panel del proyecto

La pestaña de vista Proyecto del panel del proyecto muestra el contenido del proyecto activo. Consta de los siguientes elementos:

- Una pequeña barra de herramientas con cuatro botones:
 - Icono Cerrar proyecto: cierra el proyecto activo.
 - Añadir archivos/paquetes/clases: se abre el cuadro de diálogo Añadir al proyecto.
 - Eliminar del proyecto: elimina del proyecto los archivos seleccionados.
 - Actualizar: actualiza el proyecto y busca automáticamente paquetes de código (Projecto!Propiedades de proyecto!General).
- Una lista desplegable de todos los proyectos abiertos.

- Una vista en árbol con el contenido del proyecto activo.
- Un menú 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.

La pestaña de vista Buscador de archivos del panel del proyecto

La pestaña Buscador de archivos del panel del proyecto muestra los directorios inicial y del proyecto, así como las unidades del ordenador. Los iconos del directorio tienen el mismo aspecto que los del cuadro de diálogo Selección de archivos. La pestaña Buscador de archivos del panel del proyecto contiene los siguientes elementos:

- Una pequeña barra de herramientas con tres botones:
 - Icono Cerrar proyecto: cierra el proyecto activo.
 - Icono Favoritos: Añade la unidad, el directorio o la carpeta seleccionado a la parte superior de la vista en árbol del buscador de archivos, y lo identifica mediante el icono Favoritos. Añade el directorio de Favoritos al cuadro de diálogo Selección de archivos. Permite organizar la unidad, el directorio o la carpeta de Favoritos que se ha añadido.
 - Actualizar: actualiza el proyecto y busca automáticamente paquetes de código (ProyectoGeneralPropiedades de ProyectoGeneral).
- Una lista desplegable de todos los proyectos abiertos.
- Una vista en árbol de los directorios inicial y del proyecto y las unidades del ordenador.
- Un menú sensible al contexto para cada *nodo* (por ejemplo, nodos de unidad, directorio, carpeta, favoritos o archivo) del panel del proyecto, al que se puede acceder pulsando el nodo con el botón derecho del ratón.

Ejecución de aplicaciones desde el panel del proyecto

**Es una función de
JBuilder Developer y
Enterprise.**

El asistente Creador de compiladores y el menú contextual del panel del proyecto permiten ejecutar una aplicación desde el panel del proyecto. Cuando se crea un archivo JAR para una aplicación por medio del asistente del Creador de compiladores y se incluye la configuración de la clase principal se puede ejecutar el `make` y ejecutar la aplicación por medio de los comandos del menú contextual. Si desea más información, consulte [“Creación de archivos JAR ejecutables con el asistente Creador de compiladores” en la página 6-4](#).

Creación de proyectos

El Asistente para proyectos permite crear proyectos desde cero. El Asistente para Proyecto para código existente permite crear proyectos a partir de código y recursos anteriores.

Hay varias formas de crear un proyecto completamente nuevo:

- Elija Archivo|Nuevo proyecto en el menú principal.
- Seleccione Archivo|Nuevo y haga doble clic en el icono Proyecto de la ficha Proyecto de la galería de objetos.
- Pulse el botón Nuevo de la barra de herramientas principal y haga doble clic en el icono Proyecto de la ficha Proyecto de la galería de objetos.
- Haga clic a la derecha del botón Nuevo para acceder al menú desplegable y elija Proyecto|Proyecto.



Si desea crear un proyecto a partir de código y recursos ya creados, haga doble clic en el icono Proyecto para código existente de la ficha Proyecto de la galería de objetos. También puede abrir el menú desplegable del botón Nuevo de la barra de herramientas y elija Proyecto|Proyecto para código existente.

Adición de archivos, clases, paquetes o aplicaciones a un proyecto

Las funciones disponibles varían según la edición de JBuilder.

El IDE de JBuilder proporciona numerosas opciones para añadir archivos y paquetes a un proyecto: se pueden utilizar la barra de herramientas del panel del proyecto, el menú principal y el menú contextual del panel del proyecto. Puede añadir aplicaciones a un proyecto mediante el menú principal. Para completar la tarea del proyecto, consulte los siguientes temas.

- [“Creación de archivos” en la página 4-13](#)
- [“Adición de archivos vacíos” en la página 4-14](#)
- [“Adición de aplicaciones” en la página 4-14](#)
- [“Adición de archivos existentes” en la página 4-14](#)
- [“Adición de archivos, carpetas, directorios o paquetes” en la página 4-14](#)
- [“Creación de JAR ejecutables para aplicaciones” en la página 4-16](#)
- [“Creación de archivos y paquetes para aplicaciones web” en la página 4-16](#)

Creación de archivos

Para crear un nuevo archivo fuente de clases Java desde el menú principal:

- 1 Elija Archivo|Nueva clase en el menú principal.
Se abre el Asistente para clases.
- 2 Rellene los campos para paquete, nombre de clase y clase base.
- 3 Marque las opciones para crear elementos de clase.

Para acceder a otro método rápido de creación automática de clases, haga clic con el botón derecho del ratón en un proyecto, un paquete o un nodo de archivo, en el panel del proyecto, y elija **NuevoClase**. Se abre el cuadro de diálogo **Nueva clase**, que incluye opciones para la definición del nombre de paquete y clase.

Adición de archivos vacíos

Para añadir un archivo vacío desde el menú principal:

- 1 Seleccione **Archivo** | **Archivo nuevo**.
Se abre el cuadro de diálogo **Crear archivo**.
- 2 Rellene los campos **Nombre** y **Directorio**.
- 3 Elija un tipo de archivo en la lista desplegable.
- 4 Marque la opción **Añadir archivo al proyecto**.
(Para asegurarse de que el nuevo archivo se añade al proyecto).

Adición de aplicaciones

Para añadir una nueva aplicación desde el menú principal:

- 1 Seleccione **Archivo** | **Nuevo**.
- 2 Haga doble clic en el icono **Aplicación** de la ficha **General** de la galería de objetos.
Se abre el **Asistente para aplicaciones**.
- 3 Lleve a cabo los pasos del **Asistente para aplicaciones**.

Adición de archivos existentes

Existen varias maneras de añadir archivos existentes al proyecto:



- Pulse el botón **Añadir archivos/paquetes/clases** de la barra de herramientas del panel del proyecto y seleccione un archivo o un paquete.
- Seleccione un archivo del buscador y arrástrelo al panel del proyecto.
- Haga clic con el botón derecho del ratón en el nodo del archivo del proyecto en el panel de proyecto y elija **Añadir archivos/paquetes/clases** en el menú contextual.

Cuando se selecciona un archivo en un visualizador o se arrastra al panel del proyecto, el archivo aparece como nodo del panel del proyecto.

Adición de archivos, carpetas, directorios o paquetes

Para añadir un archivo, carpeta, directorio o paquete desde el menú contextual del panel del proyecto:

- 1 Pulse el botón derecho del ratón en el proyecto, archivo o paquete del panel de proyecto.

**Es una función de
JBuilder Developer y
Enterprise.**

Nota La fuente del menú contextual es el paquete y cualquier archivo en la vía de acceso a fuentes del proyecto (que puede ser el dependiente del paquete, del proyecto o del directorio de navegación).

2 Seleccione Nuevo.

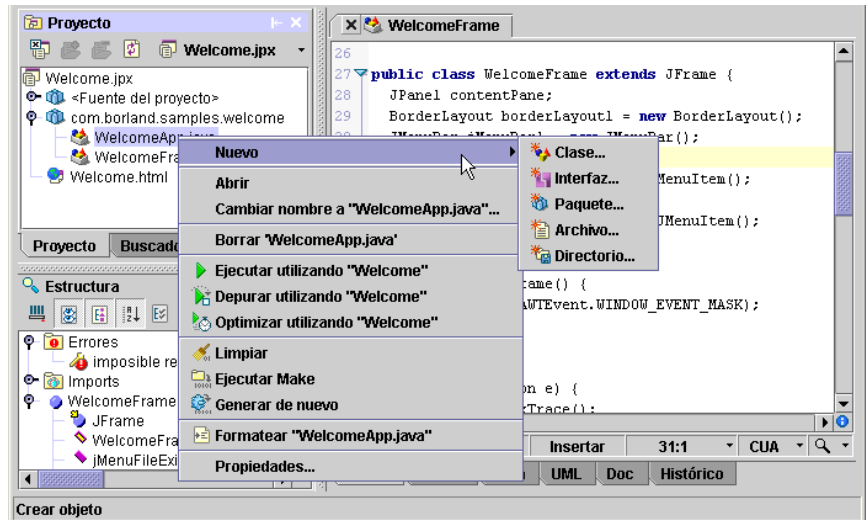
Según el archivo o paquete que seleccione, las opciones del menú incluyen:

Opciones de menú	Acción de menú
NuevoClase	Abre el Asistente para clases.
NuevoInterfaz	Abre el Asistente para interfaces.
NuevoPaquete	Abre el cuadro de diálogo Crear paquete.
NuevoArchivo	Abre el cuadro de diálogo Crear archivo.
NuevoDirectorio	Abre el cuadro de diálogo Crear directorio.
NuevoCarpeta	Abre el cuadro de diálogo Crear carpeta.
NuevoVista del directorio	Abre el cuadro de diálogo Seleccione directorio.

3 Rellene los campos del asistente o del cuadro de diálogo para el archivo o paquete.

Las opciones que aparecen en el menú contextual del panel del proyecto dependen del nodo y de la versión de JBuilder que se esté utilizando.

Figura 4.2 Menú contextual del panel de proyecto



**Es una función de
JBuilder Developer y
Enterprise.**

Creación de JAR ejecutables para aplicaciones

Si desea crear un archivo JAR que permita ejecutar una aplicación desde el panel del proyecto, consulte [“Inicio rápido por medio de los asistentes” en la página 6-4](#). Si desea más información sobre la creación de archivos JAR ejecutables por medio del Creador de compiladores, consulte “El Creador de compiladores” en *Creación de aplicaciones con JBuilder*.

Creación de archivos y paquetes para aplicaciones web

Para crear un archivo o paquete para una aplicación web (dentro del directorio o subdirectorio raíz de la aplicación web) desde el menú contextual del panel del proyecto, consulte “Directorio raíz” en la *Guía del desarrollador de aplicaciones web*. Es una función de JBuilder Enterprise.

Si desea obtener más información acerca de la creación y adición a proyectos en JBuilder, consulte el [Capítulo 7, “Proyectos de JBuilder”](#).

El panel del proyecto

Puede desplazarse libremente por el árbol del proyecto, pulsar y seleccionar varios archivos o arrastrar y colocar archivos entre los nodos antecesores del panel del proyecto. Se pueden arrastrar archivos a un paquete desde el panel del proyecto o desde el escritorio. Los nodos del paquete responden adecuadamente ante los archivos que se colocan, y los ordenan alfabéticamente.

También puede pulsar el botón derecho y seleccionar elementos en el menú contextual, sin necesidad de abrir ningún archivo. Las opciones del menú contextual varían según el tipo del archivo seleccionado. Para abrir un archivo y mostrarlo en el panel de contenido de JBuilder, selecciónelo en el panel del proyecto y pulse *Intro* o haga doble clic sobre él.

**Es una función de
JBuilder Developer y
Enterprise.**

Puede seleccionar y ampliar un archivo recopilatorio (JAR, WAR, EAR) en el panel del proyecto y ver los archivos que contiene. Si desea abrir un archivo en el panel de contenido para verlo, haga doble clic en el nodo. JBuilder utiliza el visualizador adecuado para ese tipo de archivo y lo abre. Por ejemplo, JBuilder utiliza su visualizador de imágenes para los archivos *.gif* o *.png*, y su navegador para los archivos HTML.

En el panel del proyecto se pueden abrir tantos proyectos como desee, 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*, para obtener más información sobre proyectos.

- “[Búsqueda en árboles](#)” en la [página 4-28](#) 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 y excluirlos del sistema de generación.

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

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, una carpeta Por hacer que contiene los comentarios de etiquetas `@todo` de Javadoc y una carpeta Conflictos Javadoc que contiene los conflictos del texto del Javadoc. La carpeta Conflictos Javadoc es una función de JBuilder Developer y Enterprise.

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. Además, cuando se hace clic en un elemento del panel de estructura, el editor se desplaza automáticamente y resalta el elemento correspondiente del código fuente. Si desea más información, consulte “[Búsqueda en árboles](#)” en la [página 4-28](#).

El panel de estructura ofrece funciones de ordenación y filtrado, presentación del Javadoc, desplazamiento por el código fuente y presentación de mensajes de error. Para obtener más información consulte los siguientes temas.

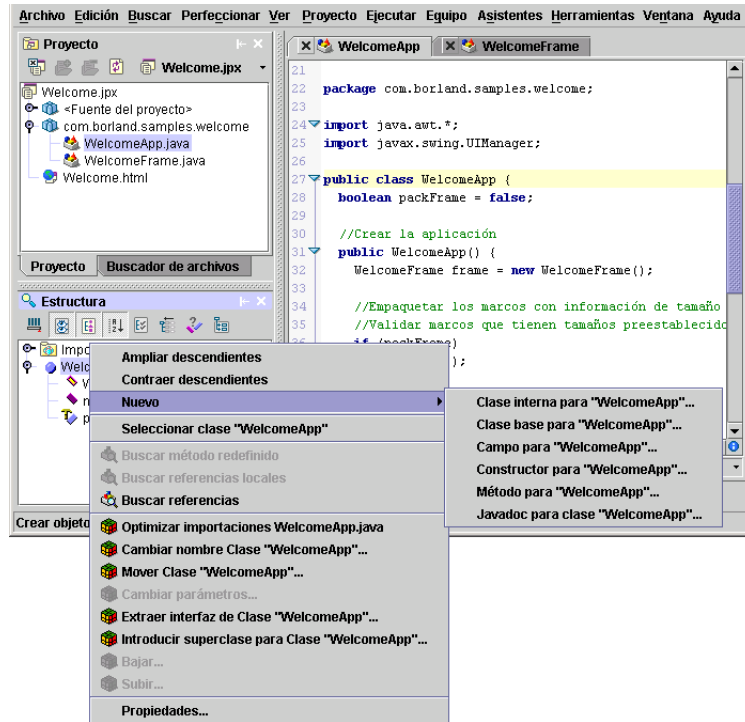
- “[Ordenación en el panel de estructura](#)” en la [página 4-17](#)
- “[Filtrado en el panel de estructura](#)” en la [página 4-19](#)
- “[Presentación de la información de Javadoc en el panel de estructura](#)” en la [página 4-20](#)
- “[El panel de estructura permite desplazarse por el código fuente](#)” en la [página 4-20](#)
- “[Presentación de los mensajes de error en el panel de estructura](#)” en la [página 4-21](#)

Ordenación en el panel de estructura

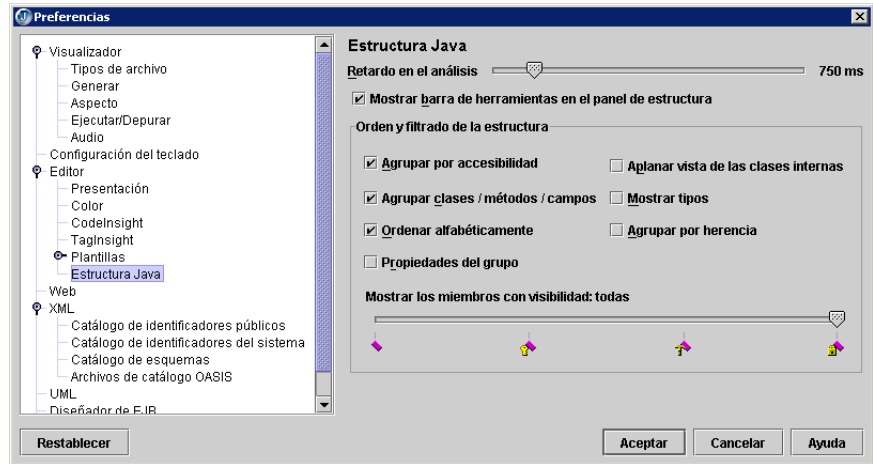
El orden, el filtrado y el retardo en el análisis del panel de estructura se pueden cambiar en el cuadro de diálogo Propiedades de visualización de la estructura. Haga clic con el botón derecho del ratón en el panel de estructura y elija Propiedades para abrir este cuadro de diálogo y modificar las opciones. También se puede ampliar y contraer el árbol del panel de estructura para facilitar la presentación y el desplazamiento. Haga clic con el botón derecho

del ratón en un elemento del panel de estructura y seleccione Ampliar descendientes o Contraer descendientes.

Figura 4.3 Menú contextual del panel de estructura











La configuración del panel de estructura también está disponible en la ficha Estructura Java del cuadro de diálogo Preferencias (Herramientas|Preferencias|Editor|Estructura Java). También se puede acceder a esta ficha desde el editor. Haga clic con el botón derecho del ratón en el panel de contenido y seleccione Opciones del editor. Si desea una descripción completa de las opciones disponibles, consulte “Preferencias de estructura de Java”, en “Personalización del editor”. Si lo prefiere, pulse el botón Ayuda de la ficha Estructura Java del cuadro de diálogo Preferencias u Opciones del editor.

Figura 4.4 La ficha Estructura Java

Filtrado en el panel de estructura

La barra de herramientas del panel de estructura muestra una variedad de iconos de filtro con los que se pueden localizar rápidamente diversos elementos de clase. Estos filtros permiten ocultar, mostrar y ordenar los datos del panel de estructura de varias formas definidas. Se pueden aplicar varios filtros, y los elementos resultantes del panel de estructura son el resultado de la suma de los filtros seleccionados. Todos los filtros se aplican por defecto, con excepción de Aplanar vista y Mostrar tipos.

Tabla 4.2 Filtros del panel de estructura

Icono	Descripción
	Filtro Ajustar accesibilidad: Abre un deslizador emergente con el que se puede ajustar la presentación de una combinación de elementos de clase públicos, protegidos, de paquete o privados en el panel de estructura.
	Agrupar por visibilidad: Agrupa los elementos de clase según la visibilidad.
	Agrupar clases/métodos/campos: Agrupa los elementos de clase según la clase, el método y el campo.
	Ordenar alfabéticamente: Ordena alfabéticamente los elementos de la a a la z.
	Propiedades del grupo: Agrupa los elementos según las propiedades individuales.
	Aplanar vista de las clases internas: Muestra todos los elementos de clase, incluidos los de las clases internas.
	Mostrar tipos: Conmuta la presentación de los tipos de devolución de los métodos y los tipos de campos.
	Agrupar por herencia: Agrupa los elementos según las clases heredadas.

Presentación de la información de Javadoc en el panel de estructura

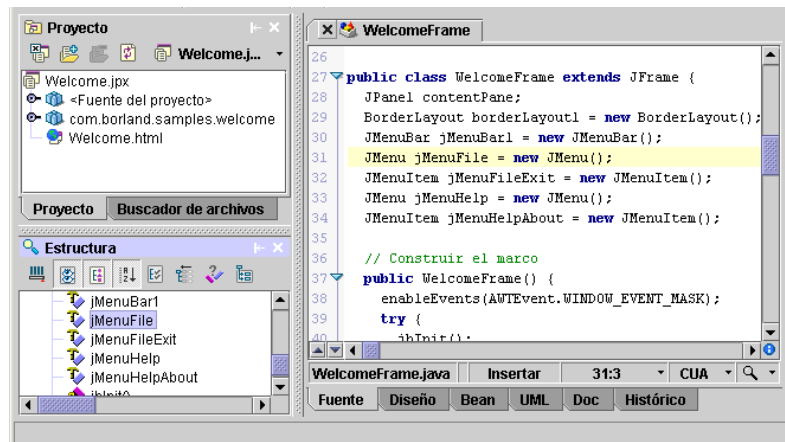
Las etiquetas `@todo` de los comentarios Javadoc también aparecen en el panel de estructura, en la carpeta Por hacer. Los conflictos Javadoc aparecen en la carpeta Conflictos Javadoc, en el panel de estructura. (Esta función pertenece a las versiones Developer y Enterprise de JBuilder.) Para corregir fácilmente los conflictos Javadoc, haga clic con el botón derecho del ratón en la carpeta Conflictos Javadoc o en el archivo Javadoc y seleccione Corregir conflictos Javadoc.

Consulte

- “Utilización de etiquetas `@todo` en el editor” en la página 5-19
- “Resolución de conflictos Javadoc” en la página 5-19. (Esta función pertenece a las versiones Developer y Enterprise de JBuilder.)
- “Conflictos en los comentarios Javadoc” en *Creación de aplicaciones con JBuilder* (Es una función de JBuilder Developer y Enterprise).

El panel de estructura permite desplazarse 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 en la que se define el elemento y la resalta.



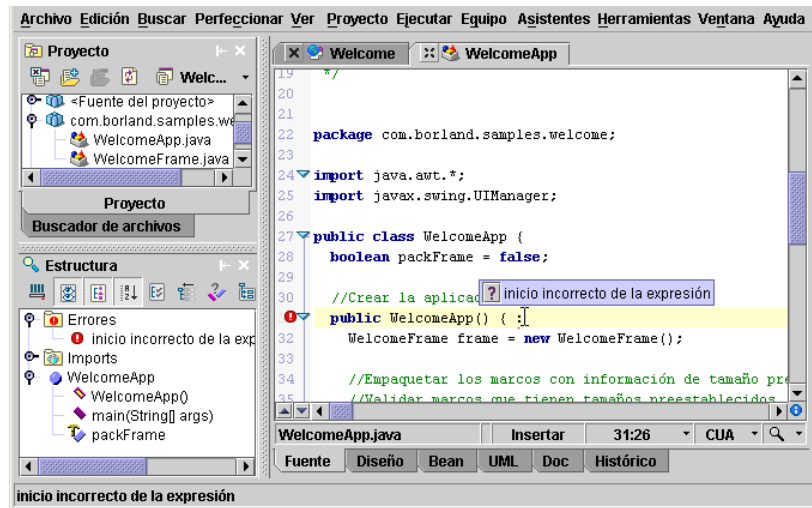
También se puede utilizar el panel de estructura para acceder a otras clases antecesoras e interfaces. Para ver el archivo `.java` de una clase antecesora o una interfaz, o el tipo de una variable que aparece en el panel de estructura, basta con hacer doble clic en el elemento o seleccionarlo y pulsar *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 los mensajes de error en el panel de estructura

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.

Los errores de sintaxis aparecen subrayados en el editor. Coloque el cursor sobre el error para ver la ayuda inmediata con el tipo del error. La ayuda inmediata cuenta con un botón con un signo de interrogación. Pulse este botón para abrir el archivo de ayuda de mensajes de error del compilador.



Consulte

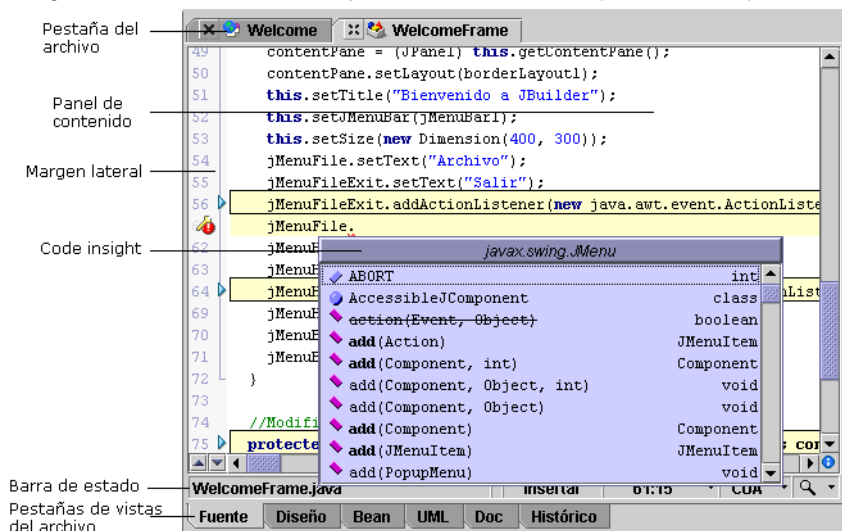
- “Mensajes de advertencia y error” en *Creación de aplicaciones con JBuilder*.
- “ErrorInsight” en la página 6-14

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. Estas pestañas ofrecen diferentes vistas, incluidas las vistas fuente, de diseño, de beans, UML, de documentos y del histórico.


Figura 4.5 Panel de contenido que muestra la vista fuente (ofrece un editor).




Pestañas de archivos del panel de contenido

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 Cerrar archivo  que tiene dos funciones:

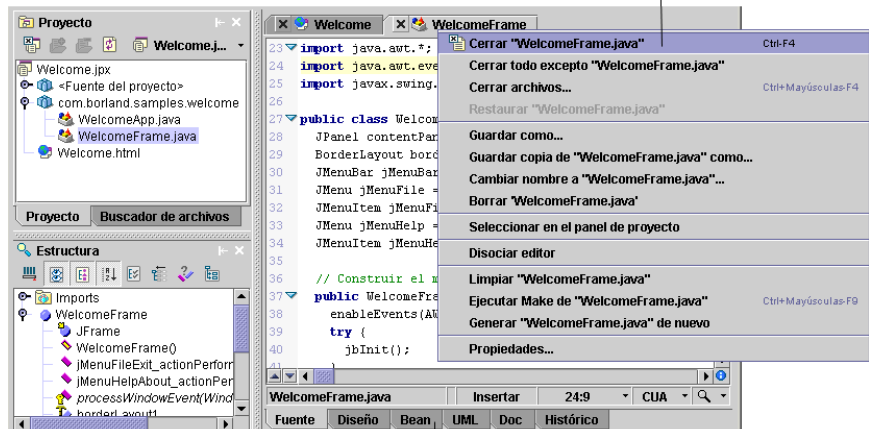
- Si se pulsa, el archivo se cierra.
- Indica si el archivo ha cambiado desde que se guardó por última vez. Si es así, tiene el siguiente aspecto:  y se pregunta si se desea guardar el archivo antes de cerrarlo.

Si un archivo es de sólo lectura, su pestaña del panel de contenido muestra un icono  que indica su estado. La pestaña de archivo muestra el nombre del archivo. La barra de título de JBuilder muestra la vía de acceso completa, el nombre y la extensión del archivo. Si se mantiene el puntero sobre la ficha de un archivo se presentan su nombre y su vía de acceso completos.

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

Desde las pestañas de archivo del panel de contenido se puede acceder a un completo menú contextual, cuyos comandos dependen del tipo de archivo. Para ver todas las opciones del menú contextual, haga clic con el botón derecho del ratón en la pestaña de un archivo del panel de contenido. El comando Seleccionar en el panel del proyecto permite desplazarse rápidamente a la ubicación del archivo abierto en el panel del proyecto.

Menú contextual de la pestaña del panel de contenidos



Pestañas de vista de archivo del panel de contenidos

Pestañas de vista de archivos del panel de contenido

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/Maximizar panel de contenidos 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

- Si desea ver otras configuraciones de los paneles, consulte el “menú Ver”.

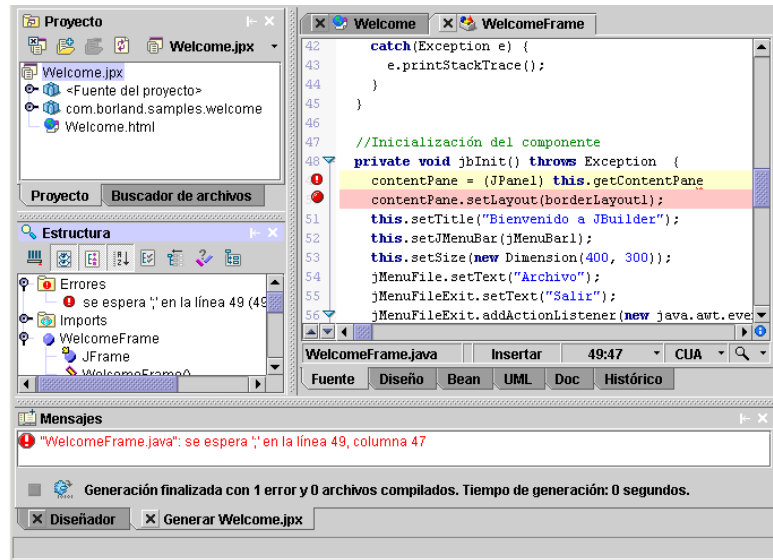
Panel de mensajes

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

- Generación y compilación
- Ejecución
- Depuración
- Búsqueda
- Perfeccionamiento de código
- Test de módulos
- 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. Durante la compilación aparece la pestaña Generar si hay errores o advertencias.

A continuación se muestra un ejemplo del IDE de JBuilder con un panel de mensajes que informa de un error de sintaxis en la parte inferior:



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 desde el 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. 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 desea, puede segregar el panel de mensajes de modo que se convierta en una ventana flotante con el fin de colocarla en cualquier parte de la pantalla.

Para segregar el panel de mensajes, haga clic en su barra de título y arrástrelo fuera del visualizador.

Para volver a integrar el panel de mensajes en la parte inferior del visualizador, pulse el icono  de la esquina superior derecha de la barra de título del panel de mensajes.

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, Intercambio inteligente,

Fijar punto de ejecución, Fuente inteligente, 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 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 inferior de la ventana del Visualizador, y le mantiene al corriente sobre las operaciones y su resultado.



La barra de estado de archivo aparece en la parte inferior de las ventanas de archivo abiertas, en el 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 en el IDE

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

Tabla 4.3 Configuraciones de teclado para desplazamientos:

Método abreviado de teclado	Acción
<i>Pestaña</i>	Permite desplazarse hacia adelante, en el orden de rotación, hasta el siguiente panel del IDE.

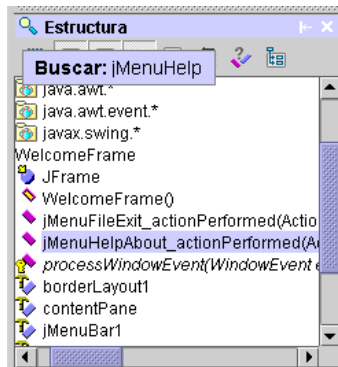
Tabla 4.3 Configuraciones de teclado para desplazamientos: (continuación)

Método abreviado de teclado	Acción
<i>Mayús+Ctrl+Tab</i>	Permite retroceder, en el orden de rotación, hasta el panel anterior del IDE.
<i>Flechas arriba y 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 las ramas de nodos de nivel superior de los árboles.
<i>Intro</i>	En el panel del proyecto, abre un archivo de código fuente seleccionado. Es equivalente a hacer doble clic. En el panel de estructura, profundiza hasta la superclase o la 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 ampliar la búsqueda a los nodos contraídos, utilice la tecla de punto (.) o Control + punto (*CTRL+.*) precedida por el nombre del nodo. El nodo se amplía y la búsqueda se limita a sus descendientes. Por ejemplo, como se muestra aquí, cuando se escribe `texteditframe.jb` en el panel de estructura se selecciona el primer componente `jButton` de la rama `TextEditFrame`.



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 *Ctrl*+(menos) o bien seleccione Buscar\Buscar clases en el menú principal para que aparezca el cuadro de diálogo Buscar clases. El cuadro de diálogo Buscar clases cuenta con dos pestañas:

En la ficha Buscar:

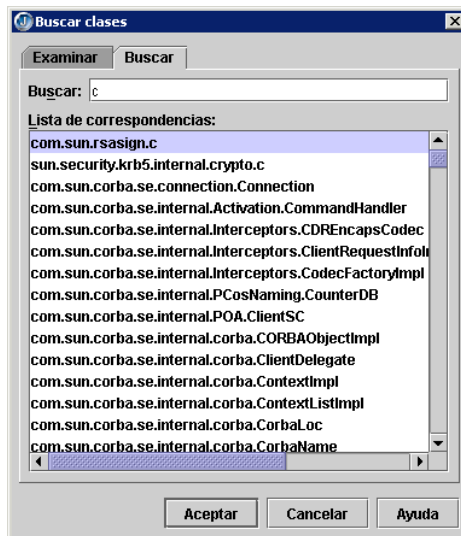
- 1 Escriba en el campo Buscar para que JBuilder rellene el campo Lista de correspondencias.

Al escribir la primera letra en el campo Buscar, aparece una lista completa de opciones de clase en 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.

Figura 4.6 Ficha Buscar



La ficha Examinar presenta los paquetes y archivos fuente utilizados en el proyecto activo. Utilice las teclas de flecha para desplazarse por el árbol. Puede también ampliar un paquete mediante el icono de expansión que se encuentra junto a él, escribiendo el nombre completo de la clase que desee abrir o escribiendo el nombre del paquete seguido de un punto (.) en el campo Nombre de clase. También puede escribir la primera letra del nombre del archivo seguida de un punto en el campo Nombre de clase; la ficha Examinar automáticamente selecciona el archivo en el árbol. La rama del árbol pertinente se amplía mientras se escribe:

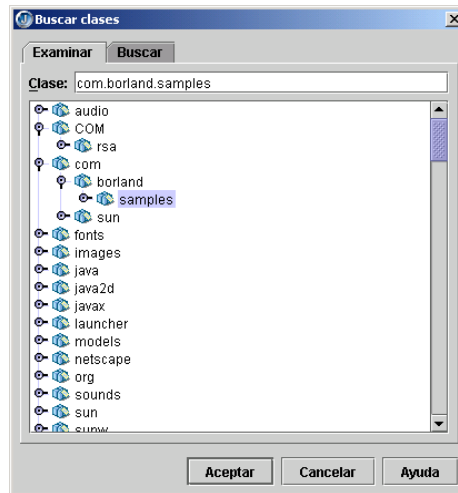
En la ficha Examinar:

- 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 ?.

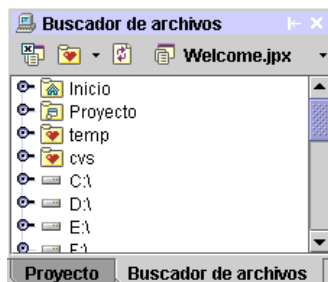
Figura 4.7 Ficha Examinar:



Utilización de Favoritos



La pestaña de vista Buscador de archivos del panel del proyecto contiene un icono Favoritos en la barra de herramientas del Visualizador de archivos. Cuando se pulsa el icono Favoritos se pueden añadir y organizar los favoritos de la vista Buscador de archivos del panel del proyecto.



Para añadir favoritos a la vista Buscador de archivos:

- 1 Seleccione la unidad, el directorio o la carpeta en el árbol de unidades y directorios.

- 2 Pulse el icono Favoritos de la barra de herramientas y elija Añadir a favoritos.

De esta forma se añade la unidad, el directorio o la carpeta seleccionado a la raíz de la vista en árbol del buscador de archivos, y lo identifica mediante el icono Favoritos. Esta acción también añade el directorio de Favoritos al cuadro de diálogo Selección de archivos.

Para organizar los favoritos en la vista Buscador de archivos:

- 1 Pulse el icono Favoritos y elija Organizar Favoritos.

Se abre el cuadro de diálogo Organizar Favoritos.

- 2 Cambie el nombre de la unidad, el directorio o la carpeta de favoritos, bórralo o desplácelo hacia arriba o abajo.

La lista Favoritos se comparte con numerosos cuadros de diálogo de selección de archivos, como Abrir archivos (al que se accede desde diversos menús y la barra de herramientas principal) y Añadir archivos o paquetes al proyecto, y contiene enlaces rápidos con los directorios favoritos. La ubicación de la lista Favoritos depende del sistema operativo. Por ejemplo, si está utilizando Windows, se encuentra en un panel a la izquierda del cuadro de diálogo Seleccione un favorito para colocar la selección del cuadro de diálogo en ese directorio.



JBuilder establece enlaces por defecto, pero, si se desea, se pueden añadir y personalizar más favoritos desde los cuadros de diálogo de selección de archivos.

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

- 1 desplácese hasta el archivo que quiere añadir.



- 2 Haga clic con el botón derecho del ratón en el icono Favoritos.

- 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 desde un cuadro de diálogo de selección de archivos:

- 1 Pulse con el botón derecho del ratón en el icono Favoritos y elija Organizar Favoritos.

Aparece el cuadro de diálogo Organizar Favoritos.

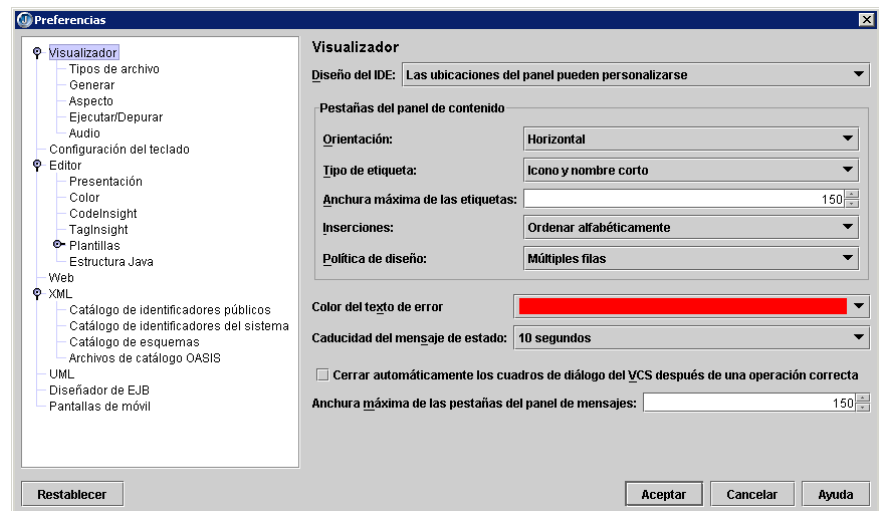
- 2 Seleccione un elemento de la lista.

- 3 Cambie el nombre del favorito, bórralo o desplácelo hacia arriba o abajo en la lista.
El orden de este cuadro de diálogo determina el de la lista Favoritos.
- 4 Pulse Aceptar o *Intro* cuando termine.

Personalización del IDE

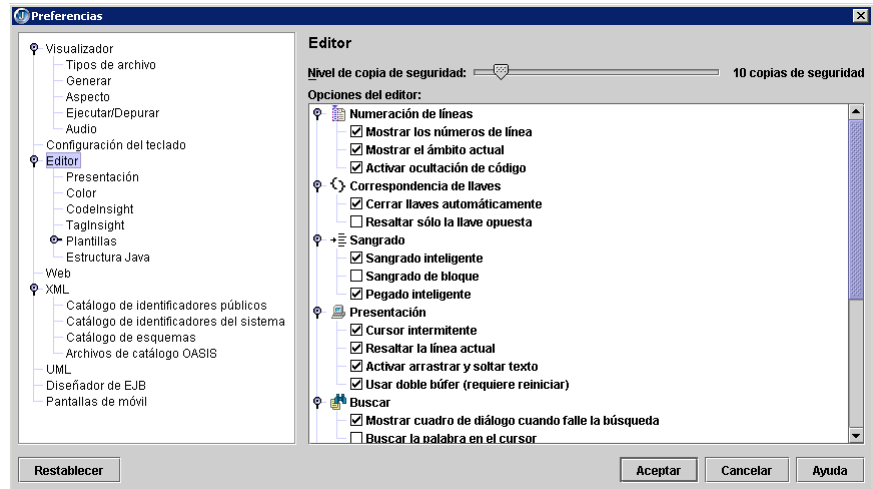
Existen diversas opciones para personalizar el aspecto y el comportamiento del IDE. Por ejemplo, es posible elegir el aspecto del Visualizador de JBuilder, seleccionar o personalizar la configuración de teclado que determina las teclas que se utilizan para acceder al IDE y especificar cómo aparecen las 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 Preferencias (Herramientas|Preferencias). Las funciones disponibles varían ligeramente según la edición de JBuilder.

Para acceder al cuadro de diálogo Preferencias, seleccione Herramientas|Preferencias.



Si desea más información sobre las diversas opciones disponibles en el cuadro de diálogo Preferencias, pulse el botón Ayuda. Si desea obtener más información acerca de la personalización del IDE, seleccione Ayuda|Entorno de JBuilder en el menú principal y pulse "Personalización del IDE de JBuilder".

Para personalizar el editor, seleccione Herramientas|Preferencias|Editor o haga clic con el botón derecho del ratón en el editor y seleccione Opciones del editor.

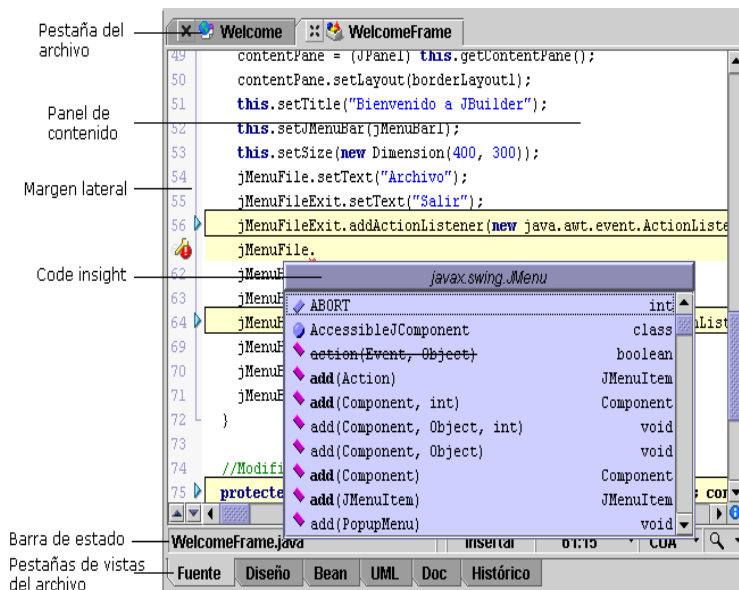


Pulse Ayuda para obtener información acerca de la utilización de las opciones del editor o bien seleccione Ayuda|Entorno de JBuilder en el menú principal y pulse "Personalización del editor".

Capítulo 5

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*. Para volver a abrir un archivo en el editor, elija *Archivo*|*Abrir de nuevo*|*Seleccionar archivo*. El cuadro de diálogo *Seleccionar archivo* proporciona una lista de los archivos que se pueden volver a abrir en el proyecto.

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. Resulta muy fácil desplazarse por el editor y buscar elementos de código haciendo clic en ellos en el panel de estructura (el que se encuentra en la esquina inferior izquierda del editor). El editor coloca automáticamente el cursor en el elemento de código, en el código fuente del editor.

El editor ofrece una amplia gama de funciones de productividad, como el resaltado de llaves correspondientes, los métodos abreviados de teclado, el resaltado de sintaxis, la personalización de la configuración de teclado del editor, los métodos abreviados de codificación (incluidos los métodos abreviados de Javadoc), la búsqueda y la impresión, así como un editor completamente personalizable. Muchas de estas funciones se pueden definir en el cuadro de diálogo *Preferencias* (*Herramientas*|*Preferencias*|*Editor*).

Existen muchas más funciones en el menú contextual del editor. Éstas varían según la edición de JBuilder que se utilice, la configuración del proyecto y la posición del cursor en el editor cuando se abre el menú. Éstas son algunas de las opciones del menú contextual: modificación, búsqueda de elementos de código y referencias, perfeccionamiento, creación de elementos de código mediante asistentes, optimización de importaciones, adición de marcadores, definición de puntos de interrupción, personalización de la vista del editor y definición de preferencias. 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.

Las pestañas de archivo del editor también ofrecen menús contextuales que varían según la edición de JBuilder, el tipo de archivo y la configuración del proyecto. Las pestañas de archivo muestran los nombres de archivo, y la barra de título de JBuilder muestra la vía de acceso completa, el nombre y la extensión del archivo abierto. Para acceder al menú contextual de las pestañas de archivo haga clic en una de ellas con el botón derecho del ratón y elija un elemento de menú.

Si desea información más detallada sobre el sistema de ayuda de JBuilder, elija uno de los siguientes temas:

- [“Personalización del editor” en la página 5-3](#) (incluidas las configuraciones del teclado)
- [“Selección, formato y búsqueda de texto” en la página 5-8](#)
- [“Búsqueda de elementos de código y definiciones” en la página 5-14](#)
- [“Aplicación de accesos directos de Javadoc” en la página 5-16](#)
- [“Definición de marcadores” en la página 5-20](#)
- [“Desplazamiento, apertura y adición de archivos” en la página 5-22](#)

- [“Métodos abreviados de codificación” en la página 5-22](#)
- [“Compatibilidad de impresión en el editor” en la página 5-23](#)

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.

Utilice el cuadro de diálogo de Preferencias para personalizar el editor. Para acceder al cuadro de diálogo seleccione **Herramientas|Preferencias** y abra la ficha **Editor**. Una forma útil de buscar las variables de código sin utilizar consiste en establecer un color de resaltado en el cuadro de diálogo **Preferencias (Herramientas|Preferencias|Editor|Color, pestaña Java)**. Cuando se trabaja con el código fuente, el resaltado desaparece automáticamente cuando las variables sin utilizar pasan a utilizarse. Si desea más información sobre las opciones disponibles, pulse el botón **Ayuda** que aparece en las distintas fichas en la categoría **Editor** del cuadro de diálogo **Preferencias**. También puede obtener más información sobre la forma de personalizar el editor si selecciona **Ayuda|Entorno de JBuilder** y selecciona **“Personalización del editor”**.

Si desea información más detallada sobre las configuraciones de teclado y la personalización de la vista del editor, elija uno de los siguientes temas:

- [“Selección de una configuración de teclado para el editor” en la página 5-3](#)
- [“Modificación de las configuraciones de teclado” en la página 5-4](#)
- [“Creación de copias de referencia de las configuraciones de teclado” en la página 5-5](#)
- [“División de la vista del código fuente” en la página 5-6](#)
- [“Presentación de los números de línea” en la página 5-7](#)

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.

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.

También puede acceder a la ficha Configuración del teclado del cuadro de diálogo Preferencias (Herramientas|Preferencias) para cambiar el esquema de configuraciones del teclado.

- 1 Abra la ficha Configuración del teclado del cuadro de diálogo Preferencias (Herramientas|Preferencias|Configuración del teclado).
- 2 En la lista Configuración del teclado elija la emulación de editor que desee.
- 3 Pulse Aceptar.

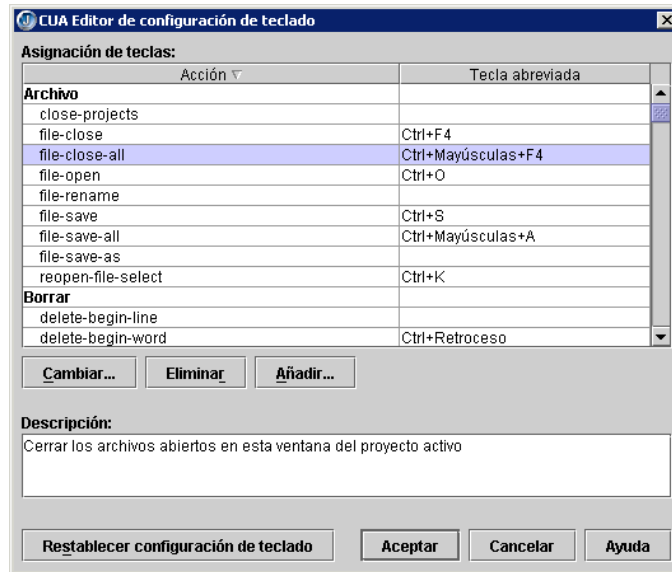
La nueva emulación de teclado se activa inmediatamente.

Modificación de las configuraciones de teclado

Puede modificar la configuración de teclado seleccionada mediante el Editor de configuración de teclado.

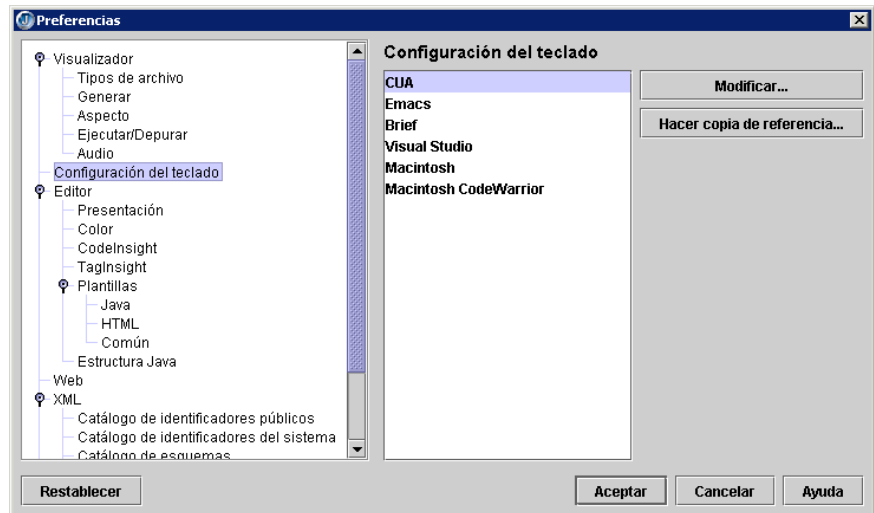
Para tener acceso al Editor de configuración de teclado:

- 1 Abra la ficha Configuración del teclado del cuadro de diálogo Preferencias (Herramientas|Preferencias|Configuración del teclado).
- 2 En la lista Configuración del teclado elija la emulación de editor que desee.
- 3 Pulse el botón Modificar contiguo a la lista Configuración del teclado.
- 4 Seleccione el comando de configuración del teclado que desee modificar o eliminar, o al que desee añadir una configuración.
- 5 Pulse el botón correspondiente a la acción que desea realizar: Cambiar, Eliminar o Añadir.
- 6 Pulse sobre las cabeceras de la tabla para cambiar el orden de clasificación.

Figura 5.1 Editor de configuración de teclado CUA

Creación de copias de referencia de las configuraciones de teclado

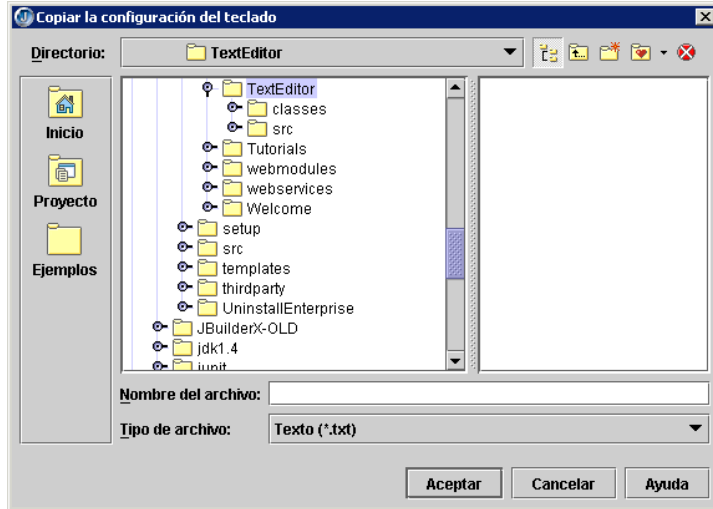
Resulta muy fácil copiar asignaciones de teclas y guardarlas con formato de archivo de texto. Se puede acceder al botón de copia de configuraciones de teclado desde la ficha Configuración del teclado del cuadro de diálogo Preferencias (Herramientas|Preferencias|Configuración del teclado).



Para copiar configuraciones de teclado:

- 1 Seleccione Herramientas|Preferencias|Configuración del teclado.
- 2 En la lista Configuración del teclado, elija la emulación que desee copiar.
- 3 Pulse el botón Hacer copia de referencia.

Se abre el cuadro de diálogo Copiar la configuración del teclado.



- 4 Seleccione el directorio en el que desea colocar la copia de referencia.
- 5 Escriba el nombre del archivo de texto en el campo Nombre de archivo.

JBuilder copia el contenido actual de la tabla en un archivo de texto y lo abre en el editor.

Si desea obtener más información sobre cómo utilizar el cuadro de diálogo Editor de configuración de teclado, haga clic sobre el botón Ayuda en el cuadro de diálogo.

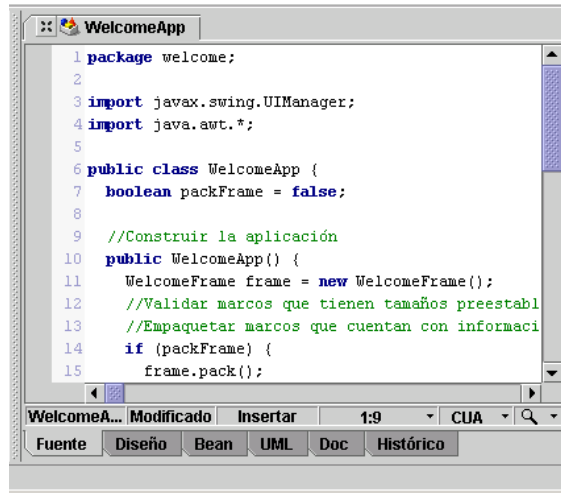
División de la vista del código fuente

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 en el 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, seleccione Vista y elija Dividir en vertical o Dividir en horizontal. Para devolver a un panel una vista dividida, haga clic con el botón derecho del ratón en una sección del panel dividido y elija Vista|Cerrar vista para cerrar la sección que tiene el foco, o Vista|Cerrar otras vistas para cerrar las otras secciones.

Presentación de los números de línea

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



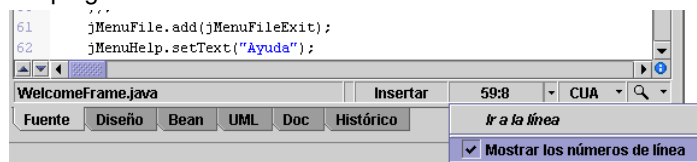
Esta función se puede activar o desactivar mediante la opción Mostrar números de línea de Numerar líneas, en la ficha Editor del cuadro de diálogo Preferencias (Herramientas/Preferencias). También es posible desplazar instantáneamente el cursor hasta un determinado número de línea. Para ello, seleccione 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.

Control 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.

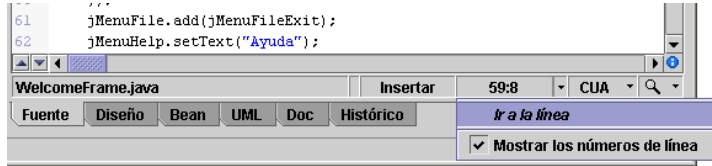


Para mostrar los números de línea, pulse la flecha abajo y active la opción Mostrar los números de línea en el menú desplegable. Para ocultarlos, pulse la flecha abajo y desactive la opción Mostrar los números de línea en el menú desplegable.



Desplazamiento a una línea determinada

Para ir a un número de línea determinado, haga clic en el número de línea o columna en el área de estado, 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, formato y búsqueda de texto

Las numerosas funciones del editor permiten seleccionar, copiar, cambiar de tamaño el texto y realizar búsquedas, así como dar formato a los elementos de código y texto cuando se está trabajando dentro del editor. Cuando se utilizan estas funciones aparecen varias funciones para completar la tarea con rapidez y eficacia. Los siguientes temas describen estas funciones del editor y proporcionan instrucciones de empleo.

- [“Selección de texto” en la página 5-8](#)
- [“Desplazamiento de texto mediante la operación de arrastrar y soltar” en la página 5-9](#)
- [“Cambio del tamaño de fuente” en la página 5-9](#)
- [“Cómo aplicar formato al código” en la página 5-10](#)
- [“Búsqueda de texto en el editor” en la página 5-13](#)

Selección de texto

Existe una gran variedad de maneras de seleccionar texto en el editor. Si el texto seleccionado incluye elementos de código duplicados, en el margen aparece el botón Sync Edit. Sync Edit es una herramienta que ayuda a modificar rápidamente los identificadores de código duplicados. Para más información consulte [“Sync Edit” en la página 6-23](#). La lista siguiente incluye varias formas de seleccionar texto en el editor.



- Seleccione una sección de texto mediante el ratón.
- Coloque el cursor en la posición inicial, mantenga pulsada la tecla *Mayús* y pulse el botón del ratón al final del área de selección.
- Mantenga pulsada la tecla *Mayús* y pulse la tecla de cursor izquierda o derecha para seleccionar texto por letras, palabras o líneas.

- Mantenga pulsada la tecla *Mayús* y pulse la tecla de cursor arriba o abajo para seleccionar texto por líneas.

Además, existen atajos de teclado en el editor que le permiten una rápida selección de líneas completas de texto mediante el número de línea que aparece en el margen lateral. Para utilizar estos métodos abreviados, asegúrese de que los números de línea aparecen en el margen lateral (consulte el apartado “[Control de los números de línea](#)” en la [página 5-7](#)).

- 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 para seleccionar un bloque de 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 para seleccionar todas las líneas de toda la fila.

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 la lista Opciones del editor, la línea recibirá automáticamente el sangrado apropiado cuando la coloque en la nueva línea. Se puede acceder a esta opción desde el cuadro de diálogo Preferencias (Herramientas|Preferencias|Editor), en la categoría Opciones de sangrado.

Cambio del tamaño de fuente

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.

Para cambiar el tamaño de fuente que se utiliza en todo el IDE, elija Herramientas|Preferencias|Visualizador|Aspecto. Seleccione un valor de la lista de desplegable Ajuste de altura. El valor por defecto es 0, que no aumenta ni disminuye el tamaño. Si se establece un valor superior, el tamaño de la fuente aumentará ligeramente respecto al valor anterior. Si se establece un valor inferior, el tamaño de la fuente se reducirá ligeramente. Por ejemplo, +2 hace

referencia a un tamaño de fuente mayor que +1 y -2 establece un tamaño de fuente menor que -1. Pruebe hasta encontrar el tamaño de fuente que le resulte más cómodo.

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

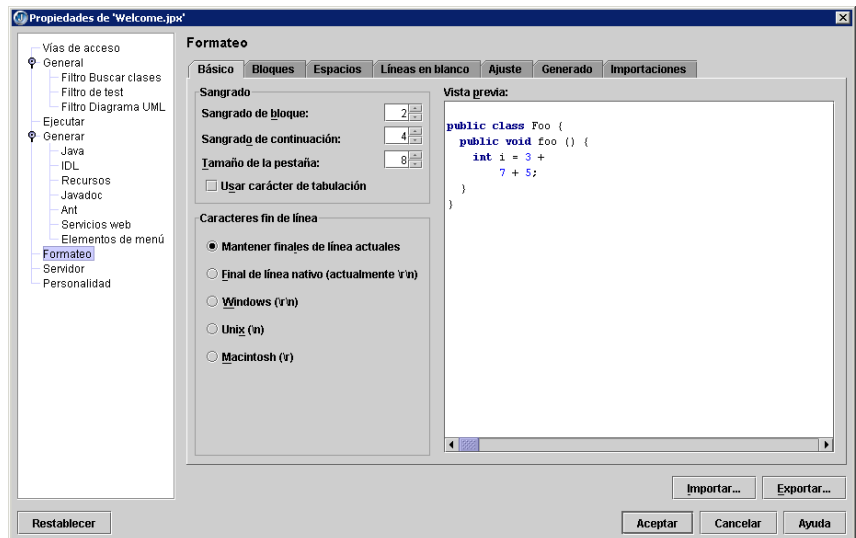
Cómo aplicar formato al código

Puede especificar sus preferencias de formato y aplicar formato automáticamente a su código fuente en el editor. Si desea obtener más información sobre el formato de código, seleccione uno de los siguientes temas:

- [“Acceso a la configuración de formato” en la página 5-10](#)
- [“Aplicación de formato al código fuente” en la página 5-11](#)
- [“Personalización de elementos de pantalla” en la página 5-12](#)
- [“Colocación de llaves en los bloques de código” en la página 5-13](#)

Acceso a la configuración de formato

La ficha Formateo del cuadro de diálogo Propiedades de proyecto ofrece diversas opciones.



Para acceder a la configuración de formato del código, elija la ficha Formateo del cuadro de diálogo Propiedades de proyecto (Proyecto|Propiedades de proyecto|Formateo). Las pestañas de esta ficha permiten personalizar el formato del código, mediante la definición de las siguientes preferencias:

- Sangrado

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

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

Nota La tecla *Tab* del teclado está configurada para el formato automático. Pulse una vez la tecla *Tab* cuando el cursor se encuentre en una línea vacía, y se desplazará automáticamente al sangrado de la línea anterior. Si pulsa de nuevo la tecla *Tab* o si la pulsa cuando el cursor se encuentra en una línea con texto, no ocurre nada. Si desea cambiar la configuración de formato automático de la tecla *Tab*, abra la ficha Configuración del teclado del cuadro de diálogo Preferencias (Herramientas|Preferencias|Configuración del teclado) o consulte [“Modificación de las configuraciones de teclado” en la página 5-4](#).

Si desea aplicar otras opciones de formato al código fuente del editor o a los paneles de contenido y mensajes, abra el cuadro de diálogo Preferencias (Herramientas|Preferencias). Es posible personalizar los siguientes elementos en la ficha Presentación del cuadro de diálogo Preferencias (Herramientas|Preferencias|Editor|Presentación).

- Márgenes del editor para texto
- Familia de fuentes del editor
- Tamaño de fuente en el editor

También es posible modificar los siguientes elementos en la ficha Visualizador del cuadro de diálogo Preferencias (Herramientas|Preferencias|Visualizador).

- Anchura máxima de las pestañas del panel de contenido
- Anchura máxima de las pestañas del panel de mensajes

Si desea más información sobre las fichas Presentación y Visualizador del cuadro de diálogo Preferencias, pulse sus botones Ayuda.

Aplicación de formato al código fuente

Para aplicar formato al código, abra un archivo en el editor y seleccione Edición|Formatear línea, pulse la tecla *Tab* o seleccione Modificar|Formatear todo. También se puede acceder a estos comandos desde el menú contextual del editor.

Importante Las propiedades de formato se aplican a todos los archivos del proyecto. También se pueden exportar las preferencias de formato de código, o importar preferencias guardadas anteriormente, mediante los botones

Importar y Exportar de 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.

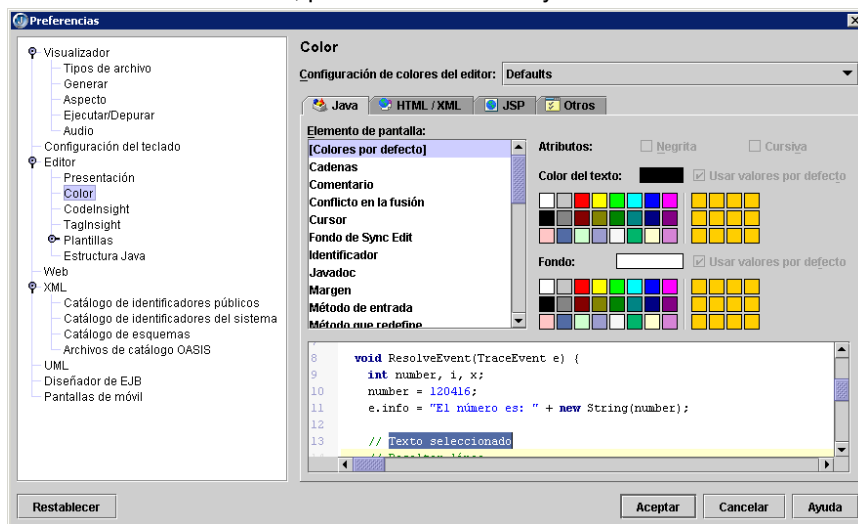
Personalización de elementos de pantalla

Utilice la ficha Color del cuadro de diálogo Preferencias (Herramientas|Preferencias|Editor|Color) para personalizar los elementos de pantalla del código en el editor. Se puede especificar el resaltado de sintaxis para cualquier elemento del cuadro de ejemplo Elementos de pantalla de la ficha Java, HTML/XML, JSP u Otros: Muchos de los elementos de pantalla del editor se pueden personalizar. Por ejemplo, se pueden definir la fuente y el color de las variables sin utilizar (lo que incluye los métodos y las importaciones) en la ficha Java. El color por defecto es el gris. También se puede personalizar el formato de los métodos que redefinen a otros y no se utilizan.

Para personalizar los elementos de pantalla:

- 1 Seleccione Herramientas|Preferencias|Editor|Color.
- 2 Elija una de las siguientes fichas: Java, HTML/XML, JSP u Otros
- 3 Elija un elemento de la lista Elemento de pantalla o haga clic en el elemento en el cuadro de ejemplo.
- 4 Configure las opciones Atributos, Color del texto y Color del fondo.

Si desea más información, pulse *F1* ó el botón Ayuda de la ficha Color.



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*.

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 *Preferencias* (*Herramientas|Preferencias|Editor*).

Tabla 5.1 Comandos de búsqueda de texto

Tarea	Comando
Buscar 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
Busca texto y lo sustituye por una cadena distinta en todos los archivos de las vías de acceso seleccionadas	Buscar Reemplazar en vía de acceso
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 el miembro de clase del archivo abierto.	Buscar Ir a miembro de clase
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
Define marcadores	Buscar Añadir marcador
Vuelve a los marcadores, los modifica o los elimina	Buscar Marcadores
Presenta una lista de los comentarios @todo de Javadoc; incluye la ubicación del archivo y del directorio	Buscar Ver los todo
Se desplaza a un elemento concreto de la lista	Ventana Histórico
Vuelve al elemento anterior de la lista	Ventana Atrás
Va al elemento siguiente de la lista	Ventana Adelante

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 elementos de código y definiciones

Cuando se trabaja en el editor se pueden buscar rápidamente elementos de código, como variables, métodos, clases y métodos que redefinen a otros, así como referencias (locales de JBuilder o de fuera de la documentación de JBuilder) de elementos de código específicos. En los siguientes temas se describe la forma de buscar referencias y elementos de código, y la forma de identificar fácilmente en el editor los elementos de código sin usar y los métodos que redefinen a otros:

Estas funciones pertenecen a las ediciones Developer y Enterprise de JBuilder.

- “Búsqueda de definiciones de símbolos” en la página 5-14
- “Búsqueda de referencias a símbolos” en la página 5-15
- “Búsqueda del método redefinido” en la página 5-15
- “Búsqueda de elementos de código y métodos que redefinen a otros” en la página 5-15

Búsqueda de definiciones de símbolos

Es una función de JBuilder Developer 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. Este comando permite desplazarse del uso de un símbolo a su definición. Si no está disponible conduce al código fuente del stub. 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`. Consulte “Configuración de teclado para las emulaciones del editor” (Ayuda\Configuraciones de teclado) para ver una lista de teclas abreviadas mediante asignación de teclas.

Nota Un símbolo de código es un fragmento de código que representa algo, como un nombre de paquete, de clase o de método.

Para acceder al comando Buscar definición, coloque el cursor en el editor y elija una de las siguientes acciones:

- Haga clic con el botón derecho en el símbolo cuya definición desea ver y elija Buscar definición.
- Coloque el cursor en el símbolo, en el código fuente, y pulse *Ctrl+Intro*.
- Coloque el cursor en el símbolo, en el código fuente, y pulse *Ctrl* mientras hace clic en el símbolo.

Nota Cuando se pulsa *Ctrl* y se hace clic en un método que redefine a otro, el editor realiza la acción Buscar definición principal. Buscar definición principal conduce al código fuente del método de redefinición actual.

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

Búsqueda de referencias a símbolos

Es una función de
JBuilder Developer y
Enterprise.

El comando Buscar referencias, disponible en el menú Buscar y en los menús contextuales del editor y de estructura, permite localizar todos los archivos fuente que utiliza un símbolo determinado. El comando Buscar referencias locales, que aparece en el menú contextual del editor, sólo muestra los archivos fuente locales que utilizan un símbolo determinado (instancias de una variable, un método, una clase o una interfaz). Los dos comandos muestran la información de referencia en el panel de mensajes, en la parte inferior del IDE de JBuilder.

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 “Descubrir referencias antes del perfeccionamiento” en “Perfeccionamiento de símbolos” en *Creación de aplicaciones con JBuilder*.

Búsqueda del método redefinido

Es una función de
JBuilder Developer y
Enterprise.

La superclase de un método se puede buscar mediante el comando Buscar método redefinido, disponible desde el menú contextual del editor o del panel de estructura. El texto de los métodos redefinidos se muestra en negrita y cursiva en el editor y en el panel de estructura.

- 1 Compile el proyecto.
- 2 Pulse el botón derecho en el método desde el editor o desde el panel de estructura.
- 3 Seleccione Buscar método redefinido.

El comando Buscar método redefinido le lleva a la superclase correcta y resalta el método que haya redefinido. Puede desplazarse por la cadena de superclases para buscar todos los métodos redefinidos de la superclase. La opción de menú Buscar método redefinido aparece en gris si el método no redefine un método de superclase.

Búsqueda de elementos de código y métodos que redefinen a otros

Es una función de
JBuilder Developer y
Enterprise.

JBuilder proporciona una forma sencilla de identificar los elementos de código sin utilizar y los métodos que redefinen a otros con una herramienta visual. Resulta fácil identificar las variables los métodos y las sentencias de importación sin utilizar, así como los métodos que redefinen a otros dentro del

código fuente, gracias al color y estilo de fuente personalizable de los elementos. El color por defecto en el editor de las variables, los métodos y las sentencias de importación sin utilizar es el gris. Por defecto, los métodos que redefinen a otros se presentan en cursiva. Esta configuración se puede personalizar en el cuadro de diálogo Preferencias.

- 1 Abra la ficha Color del cuadro de diálogo Preferencias (Herramientas|Editor|Color).
- 2 Haga clic en la pestaña Depurar.
- 3 Seleccione Variable sin utilizar o Método que redefine en el campo Elemento de pantalla.
- 4 Elija el atributo de negrita o cursiva, o un color de texto distinto, para el elemento de pantalla.

Aplicación de accesos directos de Javadoc

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 (el Asistente para Javadoc es una característica de JBuilder Developer y Enterprise). Cuando se crea el proyecto, el Javadoc se genera automáticamente.

En los siguientes temas se describen las herramientas y los métodos abreviados de Javadoc de JBuilder que se pueden personalizar y utilizar en el editor:

- [“Las plantillas de código de Javadoc” en la página 5-16](#)
- [“Adición y modificación de etiquetas Javadoc” en la página 5-17](#)
- [“JavadocInsight” en la página 5-18](#)
- [“Resolución de conflictos Javadoc” en la página 5-19.](#) (Esta función pertenece a las versiones Developer y Enterprise de JBuilder.)
- [“Utilización de etiquetas @todo en el editor” en la página 5-19](#)

Las plantillas de código de Javadoc

Con el fin de facilitar los comentarios Javadoc sobre la codificación, 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.

Para añadir rápidamente un bloque de comentarios Javadoc:

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

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 `/**` después 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) 2003</p>
 * <p>Empresa: </p>
 * @author not attributable
 * @versión 1.0
 */
```

Al escribir `/**` para la siguiente firma de método:

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

Javadoc crea el siguiente comentario Javadoc:

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

Adición y modificación de etiquetas Javadoc

El cuadro de diálogo Javadoc permite añadir y modificar etiquetas y comentarios Javadoc al código fuente Java. El cuadro de diálogo contiene los campos Descripción y Etiquetas, en los que se puede añadir y modificar texto de Javadoc. También se pueden eliminar las etiquetas, así como desplazarlas hacia arriba o abajo en la lista de etiquetas. Se puede acceder al cuadro de diálogo Javadoc desde el editor y desde el menú principal de JBuilder.

Para acceder al cuadro de diálogo Javadoc desde el editor:

- Haga clic con el botón derecho del ratón en la clase, el campo o el nombre de método, o en el comentario Javadoc.
- Seleccione **NuevoJavadoc** para método “<nombre del método>” o **Modificar Javadoc** para método “<nombre del método>”.

Para acceder al cuadro de diálogo Javadoc desde el menú principal:

- En el editor, coloque el cursor en una clase, un campo o un nombre de método, o en un bloque de comentario Javadoc.
- Seleccione Edición|Nuevo|Javadoc para método “<nombre del método>” o Edición|Modificar Javadoc para método “<nombre del método>”.

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

JavadocInsight

JavadocInsight permite introducir etiquetas Javadoc en los bloques de comentarios Javadoc. También puede aplicar al código Javadoc un color diferente del de los comentarios de bloques y líneas normales.

Para acceder a la ventana JavadocInsight y ver la lista de opciones de etiquetas Javadoc:

- Coloque el cursor en un comentario Javadoc y pulse *Ctrl+H*, *@* o *Ctrl+Barra espaciadora*.

Para definir el tiempo de demora de las ventanas emergentes de JavadocInsight:

- Seleccione Herramientas|Preferencias|Editor|CodeInsight.
Por defecto, la opción JavadocInsight automático está activada, con un retardo de 250 ms.
- Coloque el cursor en el deslizador JavadocInsight automático y mantenga pulsado el botón del ratón mientras desplaza el deslizador. También puede utilizar las teclas de cursor izquierda y derecha para mover el deslizador. Los números que aparecen a la derecha del deslizador reflejan el cambio realizado en el tiempo de demora.

Para acceder a la ficha JavadocInsight y ver las opciones de fuente y color:

- Seleccione Herramientas|Preferencias|Editor|CodeInsight.
- Pulse el botón Opciones de presentación.
Se abre el cuadro de diálogo Opciones de presentación de CodeInsight.
- Haga clic en la pestaña JavadocInsight.

**Es una función de
JBuilder Developer y
Enterprise.**

Es posible añadir etiquetas personalizadas al Javadoc. Para abrir la ficha Javadoc del cuadro de diálogo Propiedades del proyecto y poder crear, eliminar o modificar etiquetas personalizadas Javadoc, seleccione Proyecto|Propiedades de proyecto|Generar|Javadoc. Una vez que aparezca la ficha Javadoc, utilícela para llevar a cabo una de las siguientes tareas:

- Para crear una nueva etiqueta Javadoc, pulse el botón Nuevo y se abrirá el cuadro de diálogo Crear etiqueta personalizada con opciones para el

nombre, texto de cabecera y ubicación. Rellene los campos correspondientes.

- Para modificar una etiqueta Javadoc personalizada, selecciónela en la lista de etiquetas personalizadas y pulse Modificar. Modifique los campos que desee.
- Para eliminar una etiqueta Javadoc personalizada, selecciónela en la lista de etiquetas personalizadas y pulse Eliminar.

También puede utilizar opciones de la ficha Javadoc para añadir texto a la etiqueta personalizada que se muestra como una cabecera en la salida Javadoc.

Resolución de conflictos Javadoc

Es una función de
JBuilder Developer y
Enterprise.

El panel de estructura permite ver y resolver los conflictos Javadoc. Estos conflictos tienen lugar cuando las etiquetas Javadoc no coinciden con las firmas de los métodos o no contienen argumentos. Cuando se genera el proyecto, JBuilder crea la carpeta Conflictos Javadoc, en el panel de estructura. Esta carpeta contiene una lista de los conflictos Javadoc, que incluye información sobre el número de línea y el tipo de error Javadoc. Cuando se hace clic en uno de estos elementos, en la lista, el comentario Javadoc se resalta en el editor.

Para resolver un conflicto Javadoc:

- Haga clic con el botón derecho del ratón en la carpeta Conflictos Javadoc o en uno de los errores Javadoc que se muestran bajo la carpeta ampliada.
- Seleccione Corregir conflictos Javadoc.

JBuilder repara el error Javadoc del editor.

Para obtener más información sobre JavadocInsight, consulte “JavadocInsight” 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. Coloque etiquetas @todo de Javadoc dentro de los comentarios Javadoc. Estas etiquetas @todo aparecen en el panel de estructura de JBuilder en una 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> */
```

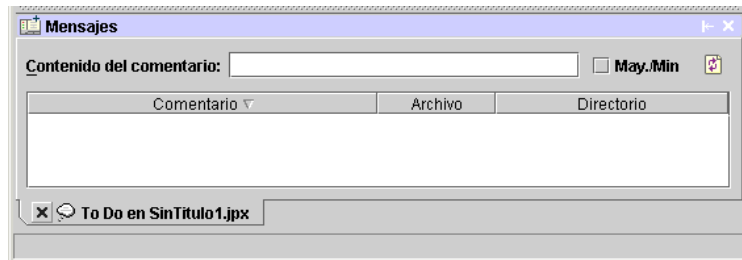
Las etiquetas @todo de Javadoc que aparecen en el Javadoc generado se personalizan en la ficha Java del cuadro de diálogo Preferencias (Herramientas|Preferencias|Editor|Plantillas|Java).

Visualización de etiquetas @todo Javadoc

Si desea ver en el panel de mensajes todos los comentarios @todo de un proyecto, un grupo de proyectos o un paquete, acceda a un comando del menú contextual del panel del proyecto o del menú Buscar del menú principal de JBuilder. Esto permite abrir rápidamente el panel de mensajes, con toda la información @todo visible.

- 1 Haga clic con el botón derecho del ratón en el nodo del proyecto o el paquete, en el panel del proyecto, y seleccione Buscar en el menú principal.
- 2 Seleccione Ver los todo

El panel de mensajes se abre en la parte inferior del IDE de JBuilder.



El panel de mensajes muestra el comentario @todo, el archivo, el directorio y una pestaña que presenta el contexto del mensaje @todo. Si desea filtrar el panel de mensajes a partir de los comentarios todo, escriba en el campo Contenido del comentario. Si el comentario distingue entre mayúsculas y minúsculas, active la casilla de selección May./Mín. El botón Actualizar permite actualizar rápidamente la lista de todo del panel de mensajes cuando se añaden otros todo. El panel de estructura actualiza automáticamente su lista de todo. También es posible ordenar el panel por nombre de archivo y directorio.

Algunos asistentes de Jbuilder, de manera opcional, generan etiquetas @todo para recordar que es necesario añadir código al stub generado por el asistente.

Consulte “Visualización de etiquetas @todo” en *Creación de aplicaciones con JBuilder* para obtener más información.

Definición de marcadores

Los marcadores permiten marcar en el editor los elementos que se desea repasar más adelante. Son persistentes y se pueden utilizar en todo el proyecto, con todos los tipos de archivo. Si desea desplazarse rápidamente al código marcado en el editor, seleccione el comando Marcadores (Buscarl Marcadores) o utilice los métodos abreviados que se pueden definir en el cuadro de diálogo Añadir marcador. El cuadro de diálogo Marcadores enumera todos los marcadores definidos en el código. Si desea más

información sobre la forma de utilizar marcadores en el editor, consulte los siguientes temas:



- [“Adición de marcadores” en la página 5-21](#)
- [“Modificación de marcadores” en la página 5-21](#)
- [“Presentación de marcadores” en la página 5-21](#)
- [“Desplazamiento a los marcadores” en la página 5-22](#)

Adición de marcadores

Es posible añadir marcadores de varias formas:

- Seleccione **Buscar** | **Añadir marcador**.
- Haga clic con el botón derecho del ratón en el elemento de código, en el editor, y seleccione **Añadir marcador**.
- Pulse *Ctrl+Mayús+M*.

Esta es la asignación de teclas del editor CUA. Si desea ver los otros editores, consulte el Editor de configuración de teclado (Herramientas | Preferencias | Configuración del teclado | Modificar).

Cuando se selecciona el comando **Añadir marcador** se abre el cuadro de diálogo del mismo nombre. Se pueden seleccionar las entradas por defecto o introducir una descripción y un método abreviado para el marcador. Cuando se termina de añadir el marcador, el icono  se muestra en el margen izquierdo del editor, junto a la selección de marcadores. Si se añade un método abreviado de teclado al marcador, el aspecto del icono  varía.

Modificación de marcadores

Puede modificar los marcadores creados:

- 1 Seleccione **Buscar** en el menú principal.
- 2 Seleccione **Marcadores**
- 3 Seleccione el marcador en el campo **Descripción**.
- 4 Pulse el botón **Modificar**.
Se abre el cuadro de diálogo **Modificar marcador**.
- 5 Rellene los campos **Descripción** y **Método abreviado**.

Presentación de marcadores

Resulta muy fácil ver la lista de marcadores y dirigirse a uno concreto en el editor:

- 1 Seleccione **Buscar** en el menú principal.
- 2 Seleccione **Marcadores**.

- 3 Seleccione el marcador en el campo Descripción.
- 4 Pulse el botón Ir a para ver en el editor el texto marcado.

Desplazamiento a los marcadores

Los siguientes métodos abreviados de teclado permiten desplazarse al marcador anterior o siguiente (con la configuración de teclado por defecto de CUA) en el editor.

- Marcador siguiente: Pulse *Ctrl+Mayús+*. (punto).
- Marcador anterior: Pulse *Ctrl+Mayús+*, (coma).

Desplazamiento, apertura y adición de archivos

Para abrir un archivo ajeno a JBuilder en el editor, puede arrastrarlo y abrirlo en el editor. El archivo debe ser de un tipo reconocido por JBuilder. Para añadir el archivo al proyecto, arrástrelo al panel del proyecto. Haga doble clic en el archivo para abrirlo en el editor.

Si desea añadir el archivo al proyecto, utilice el botón Añadir archivos/paquetes/clases que se encuentra en la parte superior del panel del proyecto o con el menú contextual del panel de proyecto, pulse el botón derecho del ratón sobre el archivo del proyecto, y seleccione Añadir archivos/paquetes/clases.

Sugerencia La función de arrastre de texto se puede desactivar en la ficha Editor del cuadro de diálogo Herramientas|Preferencias.

Métodos abreviados de codificación

JBuilder cuenta con numerosos métodos abreviados de codificación en el editor, como CodeInsight, ErrorInsight, Sync Edit y plantillas de código. CodeInsight ayuda a completar el código, mientras que ErrorInsight proporciona soluciones rápidas para errores de compilación. La herramienta Sync Edit ofrece modificación simultánea del código.

La herramienta TagInsight sirve para los archivos HTML, JSP y XML y no para los archivos fuente Java, como las otras herramientas de escritura de código. TagInsight permite completar las etiquetas de los archivos de código fuente HTML, JSP y XML. La herramienta ScopeInsight muestra el ámbito de código actual de los archivos fuente Java, además de proporcionar la ocultación del código y las flechas de desplazamiento por métodos, con las que se puede navegar rápidamente por bloques de código voluminosos. Las plantillas de código permiten insertar rápidamente en el editor elementos de código comunes para muchos tipos de archivo (los tipos por defecto son Java y HTML). Para obtener más información sobre estas características, consulte [“Los métodos abreviados de codificación” en la página 6-6](#)

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	Hora
%d	Fecha (formato largo)
%s	Fecha (formato corto)
%u	Nombre de usuario

Automatización del desarrollo de código

JBuilder cuenta con gran variedad de herramientas que aceleran el desarrollo de aplicaciones: herramientas de diseño visual para crear interfaces rápidamente, asistentes para generar código de forma automática, entre los que se encuentra el asistente Creador de compiladores, que permite crear un archivo JAR de aplicación ejecutable para una aplicación, y métodos abreviados de escritura de código y plantillas para crear y corregir código. 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. Se puede acceder a muchos de los asistentes desde la galería de objetos (Archivo|Nuevo), desde el botón Nuevo de la barra de herramientas principal y desde el menú Asistentes.



Los métodos abreviados de escritura de código incluyen CodeInsight, ScopeInsight, la ocultación de código, ErrorInsight, las plantillas de código, Sync Edit y TagInsight. Muchas de las herramientas de escritura de código se pueden personalizar en el cuadro de diálogo Preferencias (Herramientas|Preferencias) y en el cuadro de diálogo Opciones del editor (pulse con el botón derecho del ratón en el editor y seleccione Opciones del editor). El acceso a las numerosas herramientas de escritura de código es rápido y fácil. Si desea más información, consulte [“Los métodos abreviados de codificación” en la página 6-6](#).

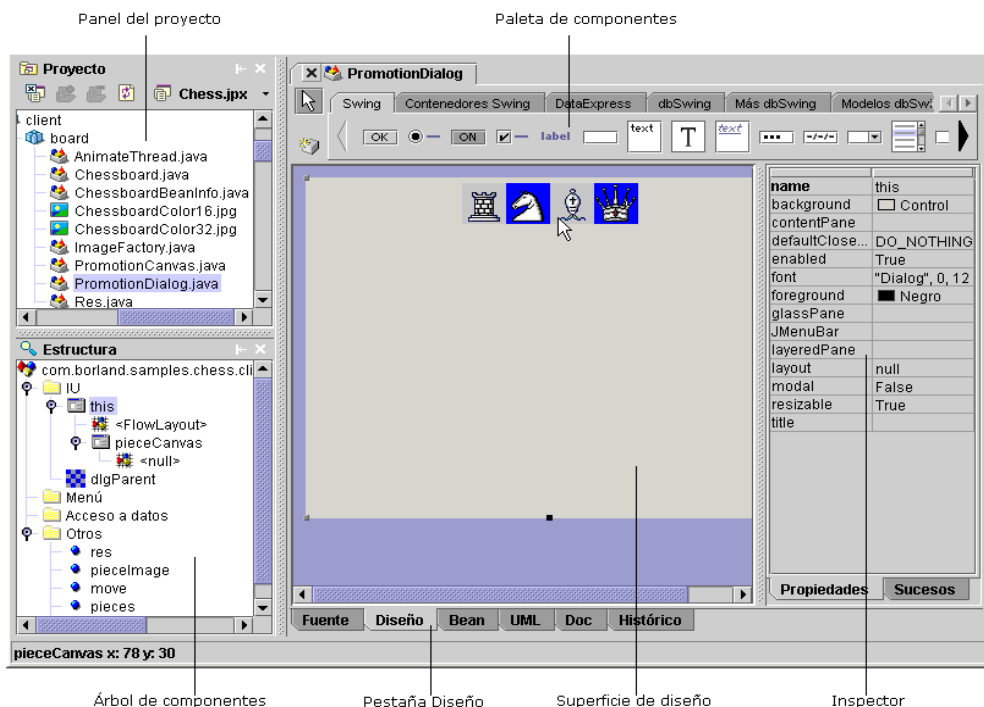
Elija entre los siguientes temas de ayuda si desea información más detallada sobre las numerosas herramientas de diseño y escritura de código y los asistentes.

- “Las herramientas de diseño visual” en la página 6-2
- “La Galería de objetos” en la página 6-3
- “Inicio rápido por medio de los asistentes” en la página 6-4
 - “Creación de archivos JAR ejecutables con el asistente Creador de compilatorios” en la página 6-4. (Esta función pertenece a las versiones Developer y Enterprise de JBuilder).
- “Creación de elementos de código” en la página 6-5
- “Los métodos abreviados de codificación” en la página 6-6

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 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*.

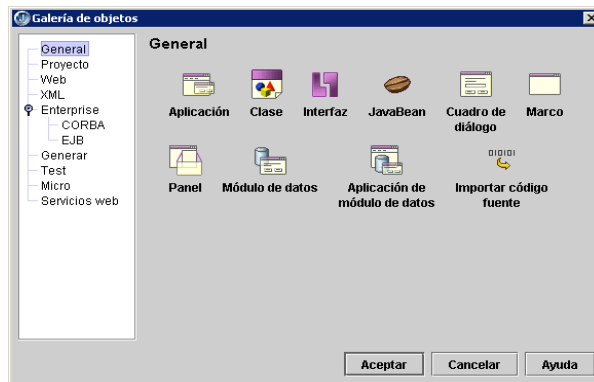
El diseñador de menús se utiliza para crear menús. Los componentes de menú aparecen en la carpeta **Menú** del árbol de componentes. Para acceder al diseñador de menús, haga clic con el botón derecho del ratón en la carpeta Menú, o en alguno de sus componentes, y seleccione Activar el diseñador.

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

La Galería de objetos

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

Figura 6.2 La Galería de objetos.



Para utilizar un asistente en la galería de objetos, elija Archivo|Nuevo o pulse el botón Nuevo de la barra de herramientas, 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. Debajo del nombre del asistente aparece un mensaje que explica el motivo por el que no está disponible. Algunos asistentes no se activan hasta que se crea o se abre un proyecto o un archivo de un tipo determinado, o bien activando un servidor 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.

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:

- ArchivolNuevo, que abre la galería de objetos.
- Menú Asistentes.
- Menú Herramientas.
- Menús contextuales de los paneles del proyecto, de estructura y de contenido (Nuevol<nombre del asistente>).



- Botón Nuevo de la barra de herramientas principal de JBuilder.

Según el nodo que se seleccione, se puede acceder a los asistentes para XML y servicios web desde el menú contextual del panel del proyecto. Los asistentes disponibles en la galería de objetos y en el menú contextual del panel del proyecto sirven para crear objetos, mientras que los demás asistentes sirven para trabajar con objetos existentes.

Los asistentes proporcionan numerosas opciones para crear rápidamente elementos de código y para realizar tareas. Es posible hacer muchas cosas, desde copiar archivos con el Asistente para importar código fuente (AsistenteslImportar código fuente o ArchivolNuevoGeneralImportar código fuente) hasta crear una interfaz que incluya comentarios Javadoc con el Asistente para interfaces (AsistenteslImplementar interfaz). El asistente Creado de compilatorios crea archivos JAR para numerosos tipos de archivo. Esto incluye un JAR ejecutable para una aplicación. Si desea más información sobre un asistente determinado, ábralo y pulse el botón Ayuda.

El Asistente para importar código fuente y la creación de archivos JAR ejecutables para aplicaciones son funciones de JBuilder Developer y Enterprise.

Creación de archivos JAR ejecutables con el asistente Creador de compilatorios

Es una función de JBuilder Developer y Enterprise.

Si desea ejecutar una aplicación haciendo clic en ella en el panel del proyecto, cree un JAR ejecutable para una aplicación mediante el Creador de compilatorios.

- 1 Elija AsistenteslCreador de compilatorios para abrir el Creador de compilatorios:
- 2 Seleccione Aplicación como Tipo de archivo en el Paso 1 del asistente.
- 3 Pulse el botón Finalizar.

Los otros botones del asistente son para configuraciones más avanzadas. Es posible especificar el archivo compilatorio, el contenido, las dependencias de bibliotecas, las opciones del descriptor y la clase principal de la aplicación.

- 4 Haga clic con el botón derecho del ratón en el nodo de proyecto del panel de proyecto.
- 5 Seleccione Ejecutar Make.
- 6 Amplíe el nodo Aplicación para mostrar el archivo JAR.
- 7 Haga clic con el botón derecho del ratón en el archivo JAR y seleccione Ejecutar utilizando valores por defecto.

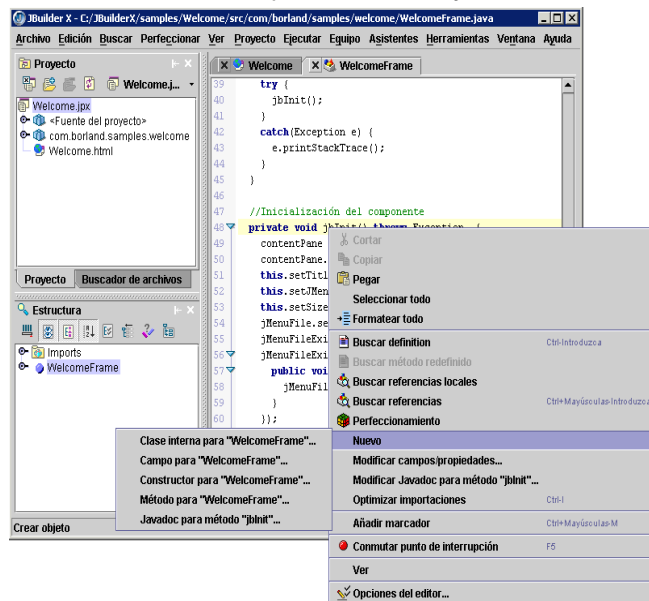
La aplicación se ejecuta.

Consulte

- “Utilización del Creador de Recopilatorios” en *Creación de aplicaciones con JBuilder*.

Creación de elementos de código

Es posible crear elementos de código para el código fuente directamente desde el editor de código fuente de JBuilder. Abra el menú contextual, haciendo con el botón derecho del ratón en el panel de contenido. Cuando se selecciona el comando Nuevo se abre el submenú Nuevo, que contiene varios comandos. Según el lugar donde se coloque el cursor cuando se hace clic con el botón derecho del ratón en el editor se tiene la posibilidad de crear métodos, constructores, campos, clases internas, clases base, Javadoc y plantillas de código. Cada opción del submenú abre un cuadro de diálogo en el que se puede crear rápidamente el elemento de código correspondiente. Si desea más información pulse el botón Ayuda del cuadro de diálogo ó F1.



Para acceder a los cuadros de diálogo en los que se pueden crear rápidamente elementos de código, haga clic con el botón derecho del ratón en el editor y elija una de las siguientes opciones.

Tabla 6.1 Comandos del submenú Nuevo (menú contextual del editor)

Comando menú	Acción
NuevolClase interna	Abre el cuadro de diálogo Nueva clase interna.
NuevolClase base	Abre el cuadro de diálogo Nueva clase base.
NuevolCampo	Abre el cuadro de diálogo Nuevo campo.
NuevolConstructor	Abre el cuadro de diálogo Nuevo constructor.
NuevolMétodo	Abre el cuadro de diálogo Nuevo método.
NuevolJavadoc	Abre el cuadro de diálogo Javadoc.
NuevolPlantilla de código	Abre el cuadro de diálogo Añadir plantilla de código (sólo se muestra cuando se selecciona un bloque de texto en el editor).

Los métodos abreviados de codificación

JBUILDER ofrece cómodas herramientas de escritura de código que proporcionan muchos métodos abreviados. No sólo existen formas rápidas de completar el código fuente; también las hay para buscar elementos del código y corregir errores. Se puede acceder a las siguientes herramientas desde el editor de JBUILDER, en el panel de contenido.

- CodeInsight proporciona listas de clases, listas de parámetros, listas de miembros y evaluación de expresiones (consulte [“CodeInsight” en la página 6-7](#)).
- ScopeInsight ofrece una herramienta de observación con la que se pueden identificar claramente las clases, métodos y expresiones asociadas en el código fuente (consulte [“ScopeInsight” en la página 6-11](#)).
- La ocultación del código permite desplazarse cómodamente por archivos fuente voluminosos (consulte [“Ocultación de código” en la página 6-13](#)).
- Las herramientas de navegación del editor proporcionan un acceso rápido a los miembros de clases (incluidos los métodos) del código fuente (consulte [“Desplazamiento a miembros de clase” en la página 6-14](#)).
- ErrorInsight ofrece un acceso rápido y una solución inmediata a los errores de codificación (consulte [“ErrorInsight” en la página 6-14](#)).
- Las plantillas de código proporcionan la posibilidad de que elementos de código que se utilizan con frecuencia se inserten rápidamente en su código (consulte [“Plantillas de código” en la página 6-17](#)).
- Sync Edit ofrece rápida modificación de código fuente y de plantillas para identificadores duplicados (consulte [“Sync Edit” en la página 6-23](#)).
- TagInsight presenta listas para completar las etiquetas de los archivos de código fuente HTML, JSP y XML (consulte [“TagInsight” en la página 6-25](#)).

Otros métodos abreviados de escritura de código incluyen la posibilidad de buscar la definición y las referencias de un símbolo y la superclase de un método redefinido. Se puede acceder a todos los métodos abreviados de búsqueda desde el menú contextual del editor. También se puede acceder a Buscar definición desde el menú Buscar y mediante una combinación de teclas, a Buscar referencias desde el menú Buscar y a Buscar método redefinido desde el menú contextual del panel de estructura.

Consulte

- [“Búsqueda de definiciones de símbolos” en la página 5-14](#)
- [“Búsqueda de referencias a símbolos” en la página 5-15](#)
- [“Búsqueda del método redefinido” en la página 5-15](#)

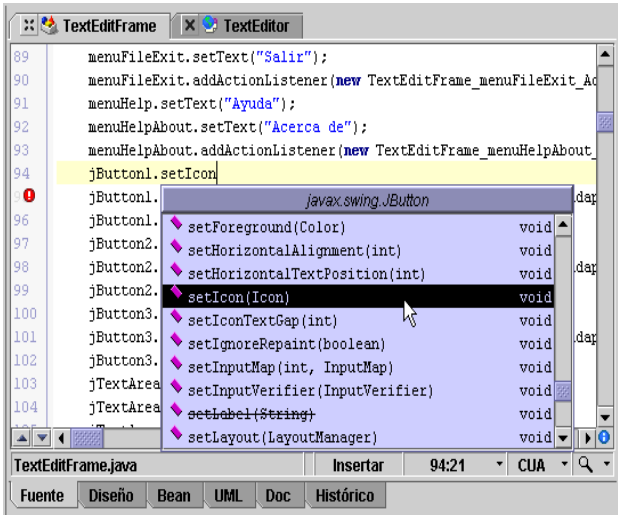
CodeInsight

CodeInsight ofrece herramientas para el rellenado del código, como listas de parámetros, clases y miembros, además de 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.
- **Evaluación de expresión de las tooltips:** los valores de las variables aparecen en el depurador. Es una función de JBuilder Developer y Enterprise.
- **ExpressionInsight:** El contenido de la expresión seleccionada en el depurador. Es una función de JBuilder Developer y Enterprise.

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

Figura 6.3 Ventana emergente de CodeInsight:



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

Para acceder a CodeInsight y a otros métodos abreviados de escritura de código, utilice estas combinaciones de teclas:

Tabla 6.2 Combinaciones de teclas

Ventana emergente	Teclas abreviadas	Acciones
MemberInsight	<i>Ctrl+H</i> o <i>Ctrl+Espacio</i>	Proporciona métodos y miembros de ámbito activo.
ParameterInsight	<i>Ctrl+Mayús+H</i>	Ofrece parámetros de métodos.
Buscar definición	<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.

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. Para abrir el Editor de configuración de teclado, seleccione Herramientas|Preferencias|Configuración del teclado, elija la configuración deseada y pulse Modificar. En JBuilder Developer 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 la selección CodeInsight, pulse *Intro*, cualquier carácter no alfanumérico ([, =, \, etc.) o la barra espaciadora

Consulte

- AyudaConfiguración del teclado
- “Personalización del editor” en ayuda en línea
- “Búsqueda de definiciones” 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+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

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 las 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

- Consulte “Optimización de las importaciones” en *Creación de aplicaciones con JBuilder*.
- “Adición y configuración de bibliotecas” en *Creación de aplicaciones con JBuilder*
- “Vías de acceso” en *Creación de aplicaciones con JBuilder*.

Opciones de inserción

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

- Consulte “Optimización de las 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.

Evaluación inmediata de expresiones

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

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* 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*

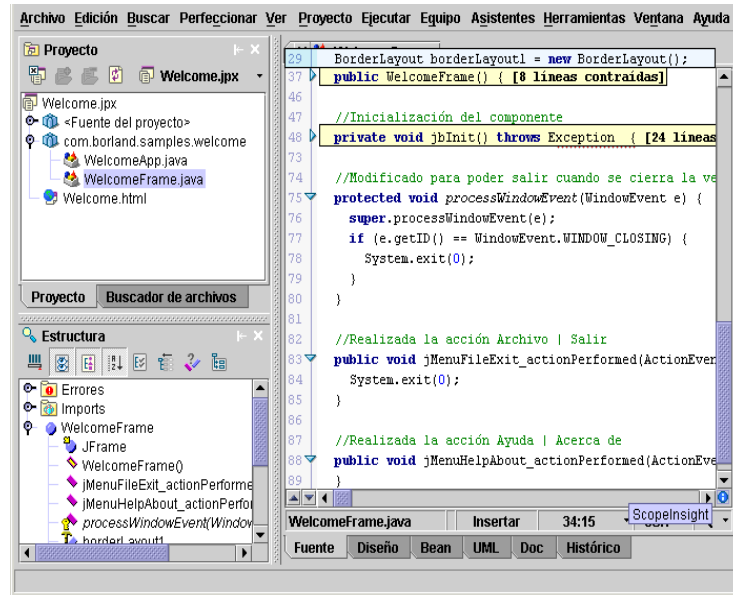
Configuración de CodeInsight

Para configurar CodeInsight, haga clic con el botón derecho del ratón en el panel de contenido, seleccione Opciones del editor y haga clic en la ficha CodeInsight, o seleccione Herramientas|Preferencias|Editor|CodeInsight. Si desea más información, pulse el botón Ayuda de la ficha CodeInsight del cuadro de diálogo.

ScopeInsight

ScopeInsight es una herramienta de observación para el código fuente en Java que identifica los niveles de anidado por encima del elemento actual y muestra el contexto del código mediante la presentación de la clase, el método o la expresión en el código fuente que tiene asociado. ScopeInsight muestra la ubicación en el archivo abierto mediante la presentación de los nombres de clase y método del método actual dentro de un cuadro emergente atenuado, en la parte superior del editor. JBuilder ofrece información de alcance adicional en el margen izquierdo del editor. El ámbito actual se

muestra con una línea continua que va del principio al fin del método, la clase o la expresión.



Si desea activar ScopesInsight para mostrar los nombres de clase y método en la parte superior del editor, pulse el botón ScopesInsight, que se encuentra en la esquina inferior derecha del editor. Cuando se coloca el puntero sobre el botón ScopesInsight aparece la ayuda inmediata ScopesInsight. Una línea continua que muestra el ámbito actual indica el lugar en que se encuentra el cursor dentro del código fuente, en el editor. La presentación del ámbito actual está activada por defecto, pero se puede desactivar.

Para activar o desactivar la presentación del ámbito actual:

- 1 Seleccione Herramientas|Preferencias|Editor.
- 2 Amplíe el nodo Margen del cuadro de lista Opciones del editor.
 - Active la casilla de selección Mostrar ámbito actual para habilitarlo.
El ámbito actual se muestra en el margen izquierdo mediante una línea continua.
 - Desactive la casilla de selección Mostrar ámbito actual para deshabilitarlo.
El ámbito actual no se muestra en el margen.

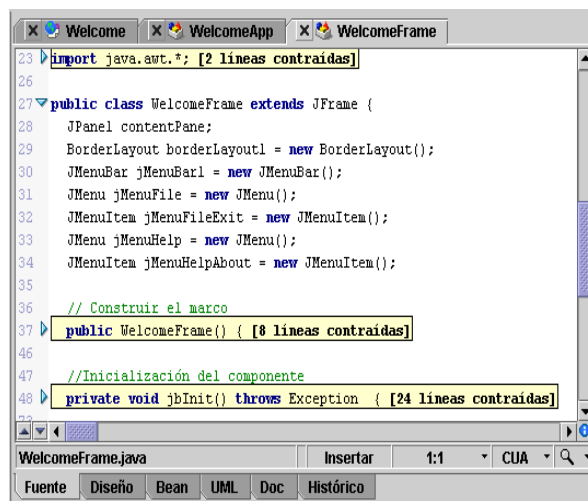
JBuilder proporciona otra herramienta visual que ayuda a localizar rápidamente los miembros del código y a identificar las variables los métodos y las sentencias de importación sin utilizar, así como los métodos que redefinen a otros dentro del código fuente Java, gracias al color y estilo de fuente personalizable. Estos elementos se personalizan en la ficha Color del cuadro de diálogo Preferencias.

- 1 Abra la ficha Color del cuadro de diálogo Preferencias (Herramientas|Preferencias|Editor|Color).
- 2 Haga clic en la pestaña Depurar.
- 3 Seleccione Método que redefine o Variable sin utilizar en el cuadro de lista Elemento de pantalla.
- 4 Asigne al texto el atributo cursiva o negrita.
- 5 Defina los colores de fuente y fondo con los que desea identificar el elemento de código.

El color por defecto para las variables sin utilizar (incluye los métodos y las sentencias de importación) es el gris, y el atributo por defecto para los métodos que redefinen a otros es la cursiva.

Ocultación de código

La ocultación de código comprime y amplía bloques de código Java, lo que facilita el desplazamiento por archivos voluminosos. La herramienta de ocultación de código permite ocultar métodos, clases y sentencias de importación. Pulse la flecha abajo contigua al código, en el margen izquierdo del editor, para ocultar un bloque de código. Cuando se oculta un bloque de código, el número de líneas contraídas aparece entre corchetes en la misma línea que la sentencia de importación, el método o la clase ocultos. Además, la flecha apunta ahora hacia la derecha. Para ampliar un bloque de código contraído, pulse la flecha que apunta hacia la derecha, en el margen izquierdo del editor.



La herramienta de ocultación de código de JBuilder está activada por defecto, pero se puede desactivar.

- 1 Seleccione Herramientas|Preferencias|Editor.

- 2 Amplíe el nodo Margen del cuadro de lista Opciones del editor.
- 3 Desactive la casilla de selección Activar ocultación de código.

Desplazamiento a miembros de clase

El editor y el IDE de JBuilder proporcionan útiles herramientas de navegación que aumentan la velocidad de escritura de código. Las flechas arriba y abajo del margen inferior izquierdo del editor permiten subir y bajar rápidamente a los métodos del código fuente. Pulse la flecha arriba para desplazarse al método anterior y la flecha abajo para desplazarse al método siguiente, dentro del código fuente. También puede utilizar el cuadro de diálogo Ir a miembro de clase, al que se accede desde el menú principal (BuscarIr a miembro de clase) para mostrar una lista de miembros de clase o buscar miembros de clase y desplazarse a ellos en el editor.

Para desplazarse a un miembro de clase por medio del menú principal:

- Seleccione BuscarIr a miembro de clase para abrir el cuadro de diálogo.
- Escriba el nombre del miembro de clase en el campo Nombre o selecciónelo en el campo Lista de miembros.
- Pulse Aceptar.

El editor muestra la ubicación del miembro de clase y resalta su línea.

ErrorInsight

ErrorInsight le facilita el acceso rápido y la solución de numerosos errores de codificación. Al introducir el código fuente en el editor, los errores de codificación aparecen de forma dinámica en el panel de estructura y también se subrayan en rojo con una línea ondulada en el editor. Los errores de codificación también aparecen en los paneles de estructura y de mensajes después de la compilación. Los iconos de ErrorInsight muestran los errores que se pueden corregir con la herramienta ErrorInsight, y aparecen junto al error de codificación en el editor y en el panel de estructura. Los iconos de errores de codificación (aquellos errores que no se pueden corregir con ErrorInsight), aparecen junto al error de codificación en el margen izquierdo del editor y en el panel de estructura. Si coloca el cursor sobre el icono de ErrorInsight en el editor, aparece el mensaje de la ayuda inmediata “Pulse *Alt + Intro* para ErrorInsight”.

Para tener acceso a ErrorInsight y corregir un error de codificación, puede:

- Pulsar el error en el panel de estructura.
- Colocar el cursor sobre el texto del código fuente en el editor y pulsar *Alt+Intro*.
- Pulsar el icono de ErrorInsight del margen izquierdo del editor que se encuentre junto al error.
- Seleccionar EditarErrorInsight en el menú principal.



Los comandos de menú dependen del error que haya seleccionado para corregir. Elija el comando que mejor se adapte a la solución de su error.

A continuación se muestra un ejemplo de las opciones de menú disponibles para corregir el error de codificación Type not found.

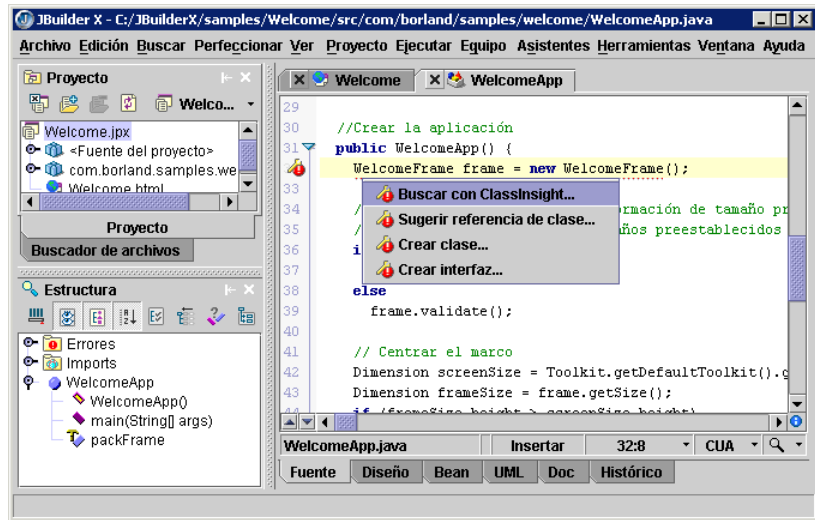


Tabla 6.3 Opciones del menú ErrorInsight

Error de código	Opciones de menú	Acción de menú
Class name or file name mismatch	Cambiar nombre a la clase	Cambia el nombre de la clase para que coincida con el nombre del archivo.
	Cambiar nombre a archivo	Cambia el nombre del archivo para que coincida con el nombre de la clase.
Class not found in import	Eliminar import	Elimina la importación del código fuente.
	Crear clase	Abre el Asistente para clases.
	Crear interfaz	Abre el Asistente para interfaces.
Class should be declared abstract	Ejecutar Make de clase abstracta	Declara abstracta la clase en el código fuente.
	Implementar métodos	Añade los métodos necesarios al código fuente.
Constructor not found	Crear constructor	Crea un constructor en el código fuente.
	Redefinir constructor	Redefine el constructor en el código fuente.

Tabla 6.3 Opciones del menú ErrorInsight (continuación)

Error de código	Opciones de menú	Acción de menú
Exception not caught	Insertar en sentencia try/catch	Inserta la excepción en una sentencia try/catch.
	Añadir cláusula Throws	Añade una cláusula throws al código fuente.
Method not found	Crear método	Abre el cuadro de diálogo Crear método.
	Sugerir referencia de método	Abre el cuadro de diálogo Seleccionar método.
Package or directory name mismatch	Cambiar la declaración del paquete	Cambia la declaración del paquete para que coincida con el código fuente.
	Mover archivo	Mueve el archivo a la carpeta correcta.
Package not found in import	Eliminar import	Elimina la sentencia de importación del código fuente.
	Crear paquete	Abre el Asistente para creación de paquetes.
Type mismatch	Añadir modificación de tipo	Inserta una conversión del tipo correcto.
Type not found	Buscar con ClassInsight	Abre el cuadro de diálogo ClassInsight.
	Sugerir referencia de clase	Abre el cuadro de diálogo Sugerir corrección con una lista de nombres de tipos parecidos.
	Crear clase	Abre el Asistente para clases.
	Crear interfaz	Abre el Asistente para interfaces.
Variable not found	Crear campo	Abre el cuadro de diálogo Nuevo campo.
	Crear variable	Abre el cuadro de diálogo Crear variable.
	Sugerir referencia de variable	Abre el cuadro de diálogo Sugerir nombre de variable con una lista de los nombres de variables parecidos.

Si selecciona un comando de menú que abre un cuadro de diálogo o un asistente, rellene los campos necesarios y pulse Aceptar. Si desea más información pulse el botón Ayuda del asistente o del cuadro de diálogo.

Plantillas de código

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 codificación. Hay plantillas de código disponibles para Java y HTML por defecto; las plantillas Java que incluyen definiciones de clase también incluyen sentencias de importación. Se puede añadir una plantilla de código de cualquier tipo en la ficha Común del cuadro de diálogo Preferencias (Herramientas|Preferencias|Editor|Plantillas|Común). La plantilla común se abre sin tener en cuenta el tipo de archivo o la ubicación. Las plantillas presentadas están determinadas por el tipo de archivo en la edición, excepto las plantillas comunes.

También se pueden definir plantillas para un tipo de archivo determinado con el fin de crear en primer lugar un grupo de plantillas, mediante la ficha Plantillas (Herramientas|Preferencias|Editor|Plantillas) del cuadro de diálogo Preferencias. Al insertar una plantilla en el código, la función de plantilla de código sangra automáticamente la plantilla insertada y el código fuente. Las preferencias de las plantillas de código también se pueden definir en el cuadro de diálogo Opciones del editor (haga clic con el botón derecho del ratón en el editor y seleccione Opciones del editor). Los siguientes temas contienen instrucciones para el uso de muchas funciones de las plantillas de código que pueden aumentar la productividad.

Consúltelos si desea ayuda para la adición de plantillas al código, la adición de tipos de archivo a la lista de las plantillas, la adición de sentencias de importación al código por medio de plantillas y la adición de macros a las plantillas de código.

- [“Adición de plantillas al código mediante la escritura de su nombre” en la página 6-18](#)
- [“Adición de plantillas al código mediante el cursor” en la página 6-18](#)
- [“Adición de tipos de archivos para plantillas de código” en la página 6-18](#)
- [“Adición de sentencias de importación al código mediante la importación automática” en la página 6-19](#)
- [“Adición de macros de plantilla a las plantillas de código” en la página 6-19](#)

Si desea crear plantillas de código propias, rodear el código con plantillas, o importar u exportar plantillas de código, consulte los siguientes apartados.

- [“Creación de plantillas de código” en la página 6-20](#)
- [“Creación de plantillas de código a partir del código fuente” en la página 6-20](#)
- [“Inserción del código fuente en plantillas de código” en la página 6-21](#)
- [“Importación y exportación de plantillas de código” en la página 6-21](#)

JBuilder permite modificar las plantillas de código por defecto y añadir plantillas propias, así como crear un desplazamiento por pestañas dentro de las plantillas. En los siguientes apartados se explica la forma de hacerlo.

- “[Modificación de plantillas de código](#)” en la página 6-22
- “[Desplazamiento mediante pestañas por el bloque de plantilla de código insertado](#)” en la página 6-23

Adición de plantillas al código mediante la escritura de su nombre

Es posible añadir una plantilla de código escribiendo el nombre de la plantilla en el código. Una vez que haya añadido una plantilla a su código, se encuentra en el modo Sync Edit. Si desea más información de cómo usar Sync Edit mode, consulte “[Edición con Sync Edit](#)” en la página 6-23.

- 1 Escriba el nombre de la plantilla de código, como `classp`, en el lugar en que desea insertarla.
- 2 Pulse *Ctrl+J*.

Si lo escrito coincide totalmente o en parte con la plantilla, JBuilder inserta la plantilla de código. Además, en el caso de los archivos Java, el editor añade la sentencia de importación de la plantilla en la parte superior del archivo fuente. Si no, se abre la lista de plantillas de código.

- 3 Si se abre la lista de plantillas de código, seleccione la plantilla y pulse *Intro*.
Si desea obtener una lista completa de los nombres de las plantillas, consulte Herramientas|Preferencias|Editor|Plantillas.

Adición de plantillas al código mediante el cursor

Es posible añadir plantillas de código colocando el cursor en la parte del código en la que desea que aparezca la plantilla en el código y mediante teclas abreviadas.

- 1 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.
- 2 Elija una de la lista, desplazándose con las teclas de flecha, y pulse *Intro*.
El editor amplía la plantilla automáticamente. Además, en el caso de los archivos Java, el editor añade la sentencia de importación de la plantilla en la parte superior del archivo fuente.

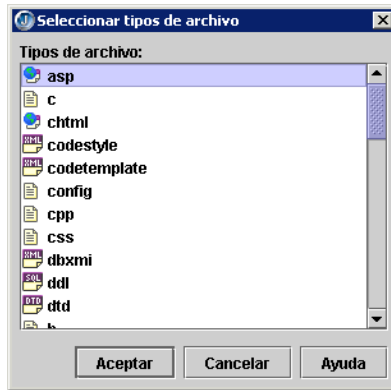
Adición de tipos de archivos para plantillas de código

La definición de tipos de archivo adicionales permite asociarlos a las plantillas de código, con el fin de ampliar sus funciones. No es necesario escribir una OpenTool adicional; basta con empezar por seguir estas instrucciones:

- 1 Seleccione Herramientas|Preferencias|Editor|Plantillas.
- 2 Pulse el botón Añadir.
Se abre el cuadro de diálogo Seleccione tipos de archivo.

- 3 Elija en la lista el tipo de archivo que desea añadir al grupo de plantillas de código.

Optativo: Abra la ficha Plantillas del cuadro de diálogo Preferencias y verá el tipo de archivo añadido en la lista Tipos de archivo con plantillas personalizadas.



Adición de sentencias de importación al código mediante la importación automática

Las plantillas de código Java incluyen una sentencia import en la parte superior del snippet de código (Herramientas|Preferencias|Editor|Plantillas). Al insertar la plantilla de código en el código fuente (*Ctrl+J*) la sentencia de importación se elimina de la plantilla y se coloca correctamente en la parte superior del código fuente.

Adición de macros de plantilla a las plantillas de código

Las macros de plantilla pueden automatizar la sustitución de nombres de identificadores en el código fuente cuando se introduce una plantilla en él. Por ejemplo, si selecciona la plantilla `equals` (que incluye la macro `$CurrentClass`) para insertarla en el código, la macro `$CurrentClass` se sustituye por el nombre de clase real de su código fuente.

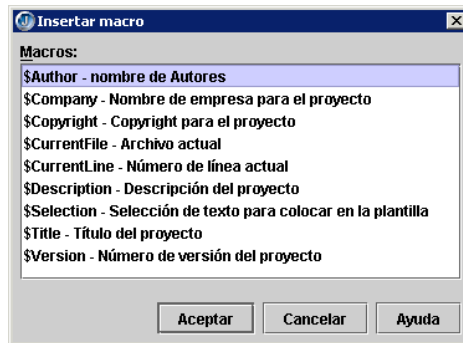
- 1 Seleccione Herramientas|Preferencias|Editor|Plantillas.
- 2 Expanda el nodo de Plantillas.
- 3 Elija uno de los nodos del nodo Plantillas; las fichas por defecto son Java, HTML y Común.

Nota El botón Insertar macro aparece atenuado en la ficha Común hasta que se añade una plantilla Común.

- 4 Coloque el cursor en el campo de texto Código en el que desee insertar la macro de plantilla.
- 5 Pulse el botón Insertar macro de la parte inferior de la ficha Plantillas.
Se abre el cuadro de diálogo Seleccione macro.

- 6 Seleccione la macro de plantilla de la lista y pulse Aceptar.

La macro de plantilla se inserta en la plantilla de código.



Creación de plantillas de código

Puede crear sus propias plantillas de código para Java, HTML y los tipos de código comunes, para sustituir los fragmentos de código que utilice con frecuencia. La creación de plantillas de código propias permite agilizar la escritura de código, con lo que la escritura repetitiva se sustituye por unas rápidas pulsaciones de teclas. Siga estas instrucciones para añadir una plantilla personalizada.

- 1 Seleccione Herramientas|Preferencias|Editor|Plantillas.
- 2 Expanda el nodo de Plantillas.
- 3 Seleccione la ficha Java, HTML o Común, según el tipo de plantilla que desea añadir.

Las plantillas comunes incluyen todos los tipos de plantillas que se muestran siempre, independientemente del tipo de archivo y de la posición dentro del archivo.

- 4 Pulse el botón Añadir.
- 5 Escriba el nombre, la descripción y el código de la nueva plantilla de código y pulse Aceptar.
Recuerde incluir la sentencia de importación adecuada.
- 6 Añada y modifique otras plantillas y, cuando termine, pulse Aceptar para cerrar el cuadro de diálogo Preferencias.

Creación de plantillas de código a partir del código fuente

Existe una forma fácil y rápida de crear plantillas de código directamente desde el código fuente.

- 1 Seleccione en el código fuente el código que desea incluir en la nueva plantilla.
- 2 Haga clic con el botón derecho en la zona seleccionada. (El color por defecto del fondo seleccionado es el azul.)

3 Seleccione NuevoPlantilla de código.

Se abre el cuadro de diálogo Añadir plantilla de código, con el código seleccionado en el área de texto Código.

4 Escriba un nombre y una descripción para la nueva plantilla.**5** Seleccione un tipo de archivo para la plantilla. Las opciones por defecto son Java y HTML.**6** Pulse el botón Insertar macro si desea añadir una macro a la nueva plantilla.**7** Pulse Aceptar para añadir la nueva plantilla a la lista.**Nota**

Para ver la plantilla de código nueva, seleccione Herramientas|Preferencias|Editor|Plantillas, amplíe el nodo Plantillas y elija la ficha correspondiente al tipo de plantilla que ha creado (Java, HTML o Común).

Inserción del código fuente en plantillas de código

Es posible insertar el código en una plantilla, en vez de insertar las plantillas dentro del código. Por ejemplo, se puede insertar en la plantilla de código `ifb` (sentencia `if`) un fragmento del código adecuado para la sentencia.

```
if () {
    Selección
}
```

1 Seleccione el código que desea insertar en una plantilla.

Puede comenzar la selección en el margen izquierdo y desplazar el cursor hacia abajo hasta la última línea de código que desee incluir.

2 Pulse *Ctrl+J*.

Se abre la lista de plantillas de código.

3 Haga doble clic en la plantilla de código desde la lista de plantillas de código.

Su código ya se ha insertado en la plantilla de código, sin que se borre y con el sangrado correcto.

Sugerencia

Después de ampliar la plantilla, utilice CodeInsight como ayuda para escribir el código. Consulte [“CodeInsight” en la página 6-7](#).

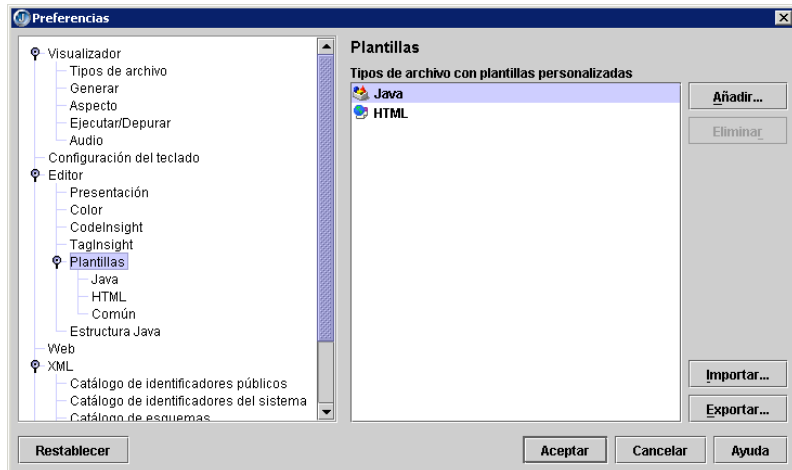
Las plantillas de código predefinidas de JBuilder tienen el formato elegido en las preferencias del proyecto. Las opciones Sangrado de bloque y Llaves se definen en las fichas Básico y Bloques del cuadro de diálogo Propiedades de proyecto|Formateo.

Importación y exportación de plantillas de código

Las plantillas de código se pueden importar y exportar como archivos XML. Se accede al cuadro de diálogo para la importación y la exportación de plantillas de código de la siguiente manera:

1 Seleccione Herramientas|Preferencias|Editor|Plantillas.

- 2 Pulse el botón Importar o Exportar de la ficha Plantillas.
- 3 Busque el directorio necesario y escriba el nombre de archivo de la plantilla.



Modificación de plantillas de código

Es posible modificar las plantillas propias y las plantillas por defecto de JBuilder y borrarlas si no son necesarias.

- 1 Abra la ficha Plantillas del cuadro de diálogo Preferencias (Herramientas|Preferencias|Editor).

También se puede acceder a la ficha Plantillas desde el menú contextual del editor. Para ello, haga clic con el botón derecho del ratón y seleccione Opciones del editor.

- 2 Expanda el nodo de Plantillas.
- 3 Seleccione un nodo subordinado del nodo Plantillas, según el tipo de plantilla que desea añadir (Java, HTML o Común).
- 4 Utilice el ratón o las flechas *arriba* y *abajo* para seleccionar la plantilla de código deseada.
- 5 Pulse el botón Modificar de la ficha correspondiente (Java, HTML o Común) para cambiar el nombre o la descripción de la plantilla seleccionada.
- 6 Escriba el nombre y la descripción de la plantilla de código y pulse Aceptar.
- 7 Escriba en el cuadro de texto Código las modificaciones deseadas para el ejemplo de código.
Recuerde incluir la sentencia de importación adecuada.
- 8 Modifique las plantillas deseadas y, cuando termine, pulse Aceptar para cerrar el cuadro de diálogo Preferencias.

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 definir de forma manual.

Desplazamiento mediante pestañas por el bloque de plantilla de código insertado

Defina el desplazamiento mediante pestañas en la plantilla de código:

- 1 Seleccione Herramientas|Preferencias|Editor|Plantillas.
- 2 Expanda el nodo de Plantillas.
- 3 Seleccione un nodo subordinado del nodo Plantillas, según el tipo de plantilla que desea añadir (Java, HTML o Común).
- 4 Haga clic en el nombre de la plantilla de la lista de plantillas de código para seleccionar la que desea modificar.
- 5 Escriba una o más barras (|) en la zona de modificación de Código en la que desea colocar las pestañas.
- 6 Pulse Aceptar.

Sync Edit

Sync Edit es una herramienta de edición muy cómoda que le permite, simultáneamente, modificar identificadores duplicados en el código fuente o en las plantillas de código (Herramientas|Preferencias|Editor|Plantillas). Se entra en el modo Sync Edit automáticamente al insertar una plantilla de código en el código fuente (*Ctrl+J*), si selecciona Edición|Sync Edit en el menú principal o si selecciona el código en el editor (si el código contiene identificadores duplicados). También puede personalizar fácilmente elementos de codificación de Sync Edit. Consulte las siguientes instrucciones para la modificación y la personalización con la herramienta Sync Edit.

- “Edición con Sync Edit” en la [página 6-23](#)
- “Personalización de los elementos de codificación de Sync Edit” en la [página 6-25](#)

Edición con Sync Edit

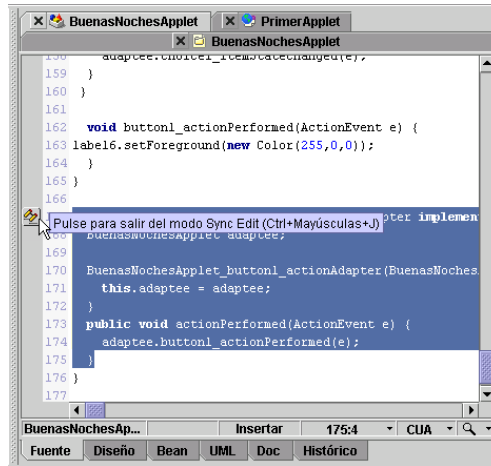


Para modificar los identificadores duplicados mediante la herramienta Sync Edit, pulse el botón Sync Edit que se encuentra en el margen izquierdo del editor al seleccionar código o al elegir Edición|Sync Edit en el menú principal. Sin embargo, si su selección de código no cuenta con identificadores duplicados, no se abrirá el modo Sync Edit. Lleve a cabo los siguientes pasos para modificar el código fuente en el editor o modificar una plantilla de código en la ficha Java, HTML o Común del cuadro de diálogo Preferencias (Herramientas|Preferencias|Editor|Plantillas):

- 1 Seleccione el bloque de código en el editor o en la plantilla de código.

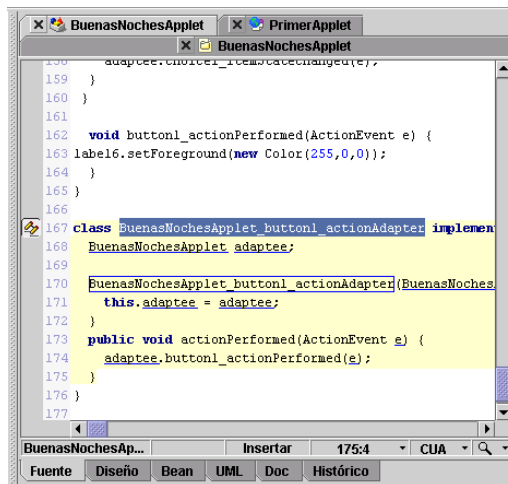


El botón Sync Edit aparece en el margen izquierdo si hay identificadores duplicados. Puede comenzar la selección en el margen izquierdo y desplazar el cursor hacia abajo hasta la última línea de código que desee incluir. El cursor vuelve automáticamente al principio de la selección para la inserción de plantillas o la modificación del código.



- 2 Pulse el botón Sync Edit del margen izquierdo del panel de contenido para entrar en el modo Sync Edit.

El primer identificador duplicado se encuentra sobre un fondo azul, los siguientes están enmarcados en un rectángulo azul. Todos los identificadores dentro de la selección están subrayados en azul.



- 3 Modifique el identificador de código duplicado que desee.

Tenga en cuenta que todos los identificadores duplicados dentro de la zona seleccionada se modifican simultáneamente. También se puede desplazar a los otros identificadores subrayados y modificar los

identificadores y sus duplicados mientras se encuentra en el modo Sync Edit.

- 4 Pulse el botón Sync Edit para salir del modo Sync Edit.

También puede salir del modo Sync Edit moviendo el cursor fuera de la zona seleccionada y haciendo clic, pulsando *Esc* o pulsando *Ctrl+Mayús+J*.

Si desea más información, pulse el botón Ayuda de la ficha Plantillas del cuadro de diálogo Herramientas|Preferencias.

Personalización de los elementos de codificación de Sync Edit

La personalización de los elementos de codificación de Sync Edit le permite mejorar la identificación de los elementos de codificación. Puede personalizar los siguientes elementos de codificación de Sync Edit: identificadores de código, fondo de Sync Edit y resaltado de Sync Edit. Se puede cambiar fácilmente el estilo o color del elemento de codificación de Sync Edit en el editor (Herramientas|Preferencias|Editor|Color).

Para personalizar los elementos de codificación de Sync Edit:

- 1 Seleccione Herramientas|Preferencias|Editor|Color.
- 2 Seleccione un elemento de codificación de Sync Edit de la lista Elementos de pantalla.

Las opciones de elementos de Sync Edit son Identificador, Fondo de Sync Edit o Resaltado de Sync Edit.

- 3 Elija entre las siguientes opciones de estilo y color:
 - Atributos: marque la casilla Negrita o Cursiva.
 - Color del texto: pulse uno de los colores para el texto o marque la casilla Utilizar valor por defecto.
 - Color del fondo: haga clic en uno de los colores para el fondo o marque la casilla Utilizar valor por defecto.

Nota puede ver los resultados de los cambios en el campo de texto que se encuentra bajo el cuadro Elementos de pantalla.

- 4 Pulse Aceptar para guardar los cambios y cerrar el cuadro de diálogo Preferencias.

TagInsight

**Es una función de
JBuilder Developer y
Enterprise.**

TagInsight proporciona herramientas para completar las etiquetas de los archivos de código fuente HTML, JSP y XML. TagInsight de JBuilder muestra automáticamente una ventana emergente sensible al tipo de archivo abierto en el editor y a la posición del cursor en el archivo. Esta ventana muestra una lista de opciones de etiqueta para el tipo de archivo correspondiente. La lista que se muestra está determinada por el contexto actual, según la posición del cursor.

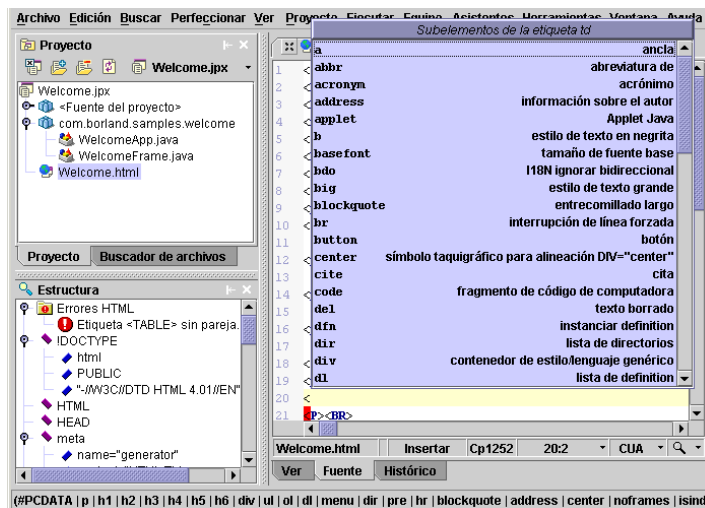
TagInsight sólo muestra en la lista las etiquetas, los elementos o los subelementos válidos. Si sólo hay un elemento válido en la posición actual del cursor no aparece ninguna lista, y TagInsight inserta el elemento en el texto. Esta función se puede activar y desactivar en la ficha TagInsight del cuadro de diálogo Preferencias (Herramientas|Preferencias|Editor|TagInsight). Cuando se selecciona una etiqueta o un elemento, TagInsight cierra la etiqueta, y cuando se selecciona un atributo, TagInsight añade un signo igual (=) seguido por comillas altas dobles, con el cursor entre ellas.

TagInsight presenta en el editor ventanas emergentes sensibles al contexto que muestran lo siguiente:

- ElementInsight: enumera los elementos disponibles.
- AttributeInsight: enumera los atributos disponibles para el elemento seleccionado.
- ValueInsight: enumera los valores enumerados disponibles para el atributo seleccionado.
- EntityInsight: enumera las entidades generales.

Para abrir TagInsight, pulse la combinación de teclas de MemberInsight de la configuración del teclado actual. Si desea ver esta combinación, abra el menú Edición y observe el atajo que se muestra junto a la opción de menú MemberInsight. Cuando se abre, TagInsight intenta abrir la ventana adecuada sensible al contexto, como ElementInsight, en la posición del cursor. Si las características automáticas emergentes están activadas en la ficha TagInsight del cuadro de diálogo Opciones del IDE (Herramientas|Opciones del IDE|Editor|TagInsight), puede introducir un corchete angular izquierdo (<) o un carácter ampersand (&) en el editor para mostrar automáticamente TagInsight. También se puede activar el temporizador introduciendo un espacio o un signo de igual (=) dentro de las etiquetas XML.

Figura 6.4 Ventana emergente de ElementInsight

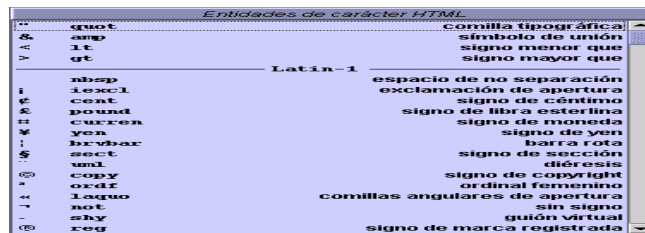


EntityInsight

TagInsight proporciona también EntityInsight, una lista de entidades para archivos HTML, JSP y XML. Para mostrar una lista de las entidades disponibles, abra TagInsight con el comando MemberInsight o escriba el carácter & en el editor si la función de apertura automática se encuentra activada. Las entidades varían según el tipo de archivo abierto en el editor. Por ejemplo, los archivos HTML contienen entidades incorporadas, de carácter y generales. Todas estas entidades empiezan con el signo &. Las entidades incorporadas incluyen & (&), <(<), >(>) y “(“). Las entidades de carácter sustituyen caracteres difíciles de escribir, como el de marca comercial (™), otros signos de puntuación, los operadores matemáticos y los caracteres no incluidos en ASCII.

Además de estas entidades, los documentos XML también pueden contener entidades generales y de parámetros. Las entidades generales se deben definir en el DTD y se pueden emplear como secciones reutilizables de texto para sustituir caracteres, texto o documentos. Las entidades generales también empiezan con el signo &. Las entidades de parámetro, que empiezan con el signo de porcentaje (%), también permiten crear secciones reutilizables de texto de sustitución, pero sólo se pueden utilizar en el DTD. TagInsight sólo acepta las entidades generales.

Figura 6.5 Ventana emergente de EntityInsight



XML TagInsight

XML TagInsight está disponible en los documentos XML definidos por un DTD o un esquema y en los archivos de generación Ant. Aunque los archivos de generación Ant no están definidos por un DTD o un esquema, TagInsight puede generar una lista de relleno que refleje Java.

Si el documento XML hace referencia a un DTD o un esquema externo situado en Internet o en una intranet, es necesario volver a hacerlo corresponder con un archivo local si se desea utilizar XML TagInsight. Después de guardar localmente los DTD y los esquemas, es necesario volver a hacer corresponder el ID con los archivos locales de una de las fichas de catálogo del cuadro de diálogo Preferencias (Herramientas|Preferencias|XML). Se puede utilizar cualquier combinación de archivos de identificadores públicos, identificadores del sistema, esquema (URI) y catálogo OASIS para obtener la correspondencia deseada. La correspondencia de esquemas URI sólo incluye los archivos XML que contienen una declaración URI de namespace que no está asociada a una ubicación de esquema; JBuilder hace corresponder el URI utilizado para identificar el namespace con un esquema.

Si desea más información sobre la correspondencia entre los identificadores y los archivos locales, pulse el botón Ayuda de una ficha del catálogo (Herramientas|Preferencias|XML).

Para hacer corresponder la ubicación del archivo local del DTD o el esquema y un catálogo, haga lo siguiente:

- 1 Guarde localmente el DTD o el esquema.
- 2 Seleccione Herramientas|Preferencias|XML y elija la ficha adecuada del catálogo, como Catálogo de identificadores del sistema.
- 3 Pulse el botón Añadir e introduzca el URI del identificador o el esquema. Por ejemplo, si la sentencia DOCTYPE tiene el siguiente aspecto:

```
<!DOCTYPE XmlEmployees SYSTEM "http://website.com/dtds/name.dtd">
```

abra la ficha Catálogo de identificadores del sistema, pulse el botón Añadir y escriba

`http://website.com/dtds/` en el campo Coincidir con el identificador del sistema.

- 4 Seleccione Archivo o Directorio, desplácese al archivo o al directorio en que se encuentra el DTD o el esquema y selecciónelo.
- 5 Pulse Aceptar.

Si desea más información sobre los catálogos XML, consulte “Definición de las preferencias de XML” en la *Guía del desarrollador de XML*.

Personalización de TagInsight

JBuilder ofrece un acceso rápido a TagInsight mediante la combinación de teclas MemberInsight (*Ctrl+Espacio* para la configuración del teclado CUA). El cuadro de diálogo Editor de configuración de teclado (Herramientas|Preferencias|Configuración del teclado|Modificar) para modificar el comando de teclado de TagInsight. La combinación de teclas de MemberInsight se encuentra en el área CodeInsight de la columna Acción del cuadro de diálogo Editor de configuración de teclado.

Las propiedades de TagInsight se pueden personalizar. Las opciones de TagInsight se pueden personalizar en las fichas Color y TagInsight del cuadro de diálogo Preferencias (Herramientas|Preferencias).

- Configure las siguientes opciones en la ficha Color del cuadro de diálogo Preferencias:
 - Modifique el color de los elementos de la pantalla.
 - Modifique los colores de texto y de fondo.
 - Cambie los atributos de la fuente.
- Configure las siguientes opciones en la ficha TagInsight del cuadro de diálogo Preferencias:
 - Active la función de apertura automática de ElementInsight, AttributeInsight, ValueInsight y EntityInsight
 - Ajuste la demora de apertura automática de TagInsight.

- Defina las opciones de relleno automático.
- Oculte y presente los elementos deseados.
- Genere los atributos obligatorios.

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.

Las propiedades de los proyectos (que incluyen las vías de acceso, los filtros, las opciones de generación y las preferencias de formato) se pueden configurar en `Proyecto\Propiedades de proyecto`. Para definir las opciones por defecto de los proyectos venideros, seleccione `Proyecto\Propiedades por defecto para proyectos`. También se pueden configurar las tecnologías que aparecen en el IDE de JBuilder (esto se denomina “configuración de personalidad”) mientras se trabaja en el proyecto. Si desea más información sobre las configuraciones de personalidad, elija `Proyecto\Propiedades de proyecto\Personalidad` y pulse el botón Ayuda.

Consulte

- “Creación y gestión de proyectos” en *Creación de aplicaciones con JBuilder*.

A continuación se facilita información más detallada sobre la creación de proyectos, su presentación y el trabajo con ellos, la adición de objetos y otros aspectos.

- “Creación de proyectos” en la página 7-2
- “Adición de archivos a un proyecto” en la página 7-4
- “Presentación de archivos” en la página 7-7
- “Adición de un archivo fuente Java” en la página 7-7
- “Eliminación y borrado de archivos” en la página 7-10
- “Cómo cambiar de nombre proyectos y archivos” en la página 7-10
- “Apertura de proyectos” en la página 7-11
- “Cierre de proyectos” en la página 7-11
- “Más información sobre los proyectos de JBuilder” en la página 7-11

Creación de proyectos

Para crear un proyecto:

- 1 Seleccione Archivo|Nuevo proyecto; o bien, seleccione Archivo|Nuevo para abrir la galería de objetos, pulse Proyecto en el panel izquierdo y haga doble clic en el icono Proyecto. Otra forma de abrir el Asistente para proyectos consiste en pulsar la flecha que se encuentra a la derecha del botón Nuevo de la barra de herramientas principal y seleccionar Proyecto en el menú desplegable.



Se abre el Asistente para proyectos:

Asistente para proyectos - Paso 1 de 3

Seleccione el nombre y la plantilla del nuevo proyecto JBuilder

Introduzca un nombre para el proyecto y la vía de acceso al directorio en que se ha de guardar. Opcionalmente, puede seleccionar un proyecto ya existente como plantilla para los valores iniciales por defecto para este asistente.

Nombre: SinTitulo1 **Tipo:** jpx ▼

Directorio: C:/jbuilder/SinTitulo1 ▼ ...

Plantilla: (Proyecto por defecto) ▼ ...

☐ Añadir proyecto al grupo activo

☐ Generar archivo de notas del proyecto

< Atrás Siguiente > Finalizar Cancelar Ayuda

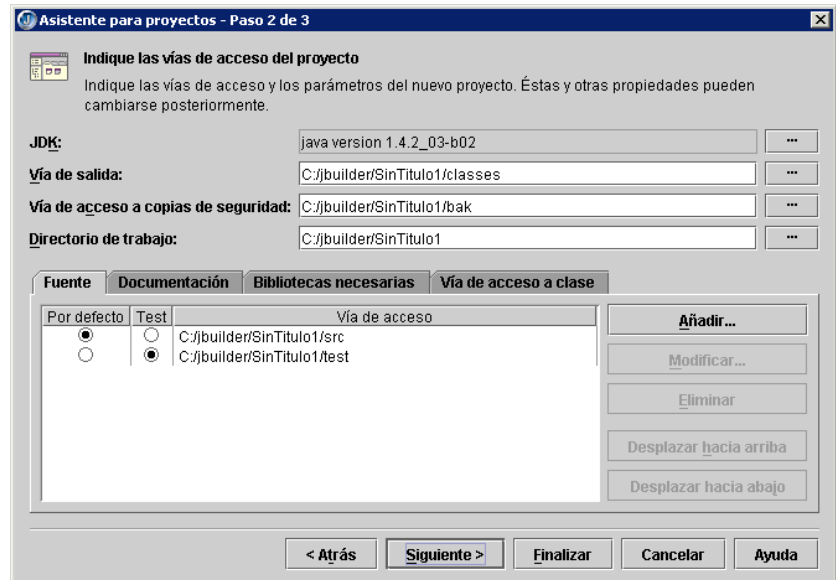
- 2 En el campo Nombre, escriba el nombre que desee dar al proyecto; por ejemplo, `MiProyecto`.

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, si desea cambiar las vías de acceso, puede hacerlo con el Asistente para proyectos, mientras crea el proyecto, o si ya está creado, con el cuadro de diálogo Propiedades de proyecto (Project Properties).

El campo JDK indica la versión del JDK que se va a utilizar para el proyecto. Pulse el botón de puntos suspensivos (...) del JDK para acceder a las opciones de cambio de versión del JDK de JBuilder.

Vía de salida especifica el lugar donde JBuilder almacena los archivos generados al generarse el proyecto.

La 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, esta ficha ofrece funciones más avanzadas que normalmente no son necesarias en los primeros proyectos. La tercera ficha incluye opciones para la escritura de código, los paquetes, las referencias y los Javadoc. Si desea más información sobre de la tercera ficha del Asistente para proyectos, ábrala y pulse el botón Ayuda.)

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

Para obtener información más detallada acerca de la creación de proyectos, consulte “Creación y gestión de proyectos” en *Creación de aplicaciones con JBuilder*. Para crear rápidamente proyectos a partir de un código anterior, consulte “Creación de proyectos a partir de archivos anteriores” en *Creación de aplicaciones con JBuilder*.

Adición de archivos a un proyecto

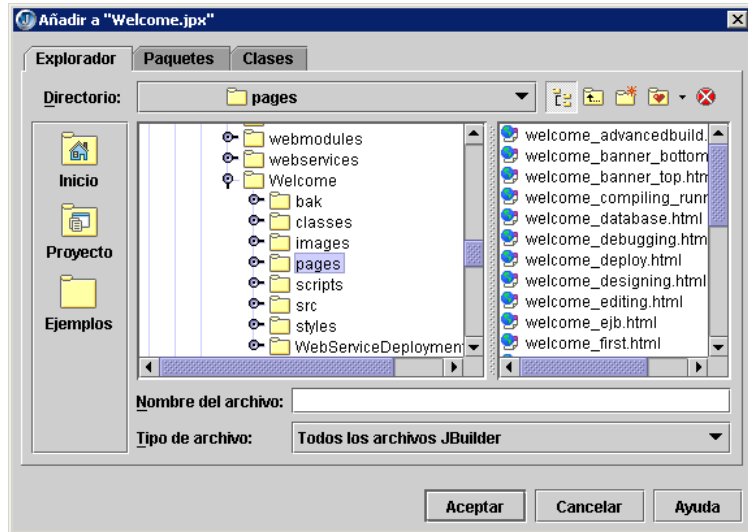
JBuilder utiliza la función de localización automática de vías de acceso a fuentes y añade las carpetas y archivos que se encuentran en el directorio fuente de su proyecto, si existen. (Puede desactivar esta función si lo desea; consulte “Recopilación automática de paquetes fuente” en *Creación de aplicaciones con JBuilder* para obtener más información).

Si esta función se encuentra desactivada, puede añadir archivos y paquetes a un nuevo proyecto mediante estos pasos:



- 1 En la barra de herramientas del panel del proyecto, pulse el botón Añadir archivos/paquetes/clases, o bien pulse el botón derecho del ratón en el nodo de archivo de proyecto del panel y, en el menú contextual, seleccione Añadir archivos/paquetes/clases.

Aparece el cuadro de diálogo Añadir a “<proyecto.jpj>”.



- 2 Seleccione la ficha Explorador para añadir un archivo, la ficha Paquetes si desea añadir un paquete o la ficha Clases para añadir una clase.
- 3 Desplácese hasta el archivo, clase o 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, clase o archivo aparece como un nodo del panel del proyecto.

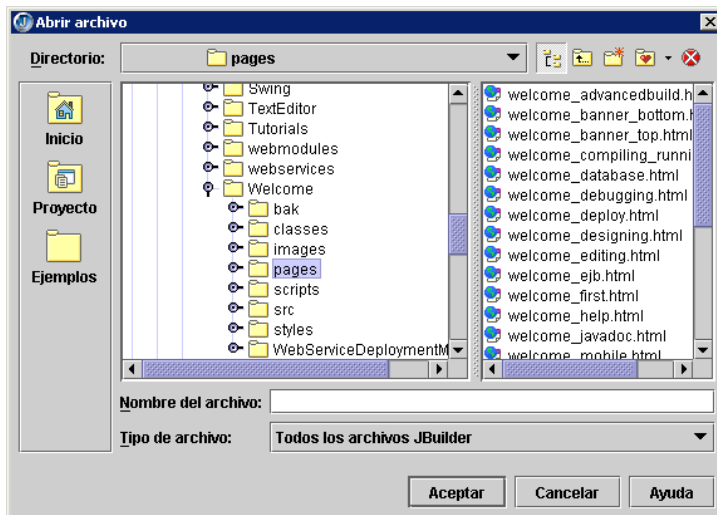
También puede utilizar el menú contextual del panel del proyecto para añadir archivos, clases o paquetes a un proyecto. Es una función de JBuilder Developer y Enterprise. Antes de utilizar el menú contextual del panel de proyecto es necesario presentar en él el archivo.

Para añadir archivos, clases o paquetes al proyecto, siga estos pasos:



- 1 Seleccione Archivo/Abrir archivo o pulse el botón Abrir archivo de la barra de herramientas principal, con el fin de abrir un archivo y mostrar su nodo en el panel del proyecto.

Se muestra el cuadro de diálogo Abrir archivo. El aspecto es parecido al del cuadro de diálogo Añadir archivos o paquetes al proyecto.



- 2 Busque el archivo en el panel izquierdo del cuadro de diálogo Abrir archivo y selecciónelo en el panel derecho.
- 3 Pulse Aceptar.
- 4 Pulse el botón derecho del ratón en el archivo o paquete del panel de proyecto.

Nota La fuente del menú contextual es el paquete y cualquier archivo en la vía de acceso a fuentes del proyecto (que puede ser el dependiente del paquete, del proyecto o del directorio de navegación).

- 5 Seleccione Nuevo.

Según el archivo o paquete que seleccione, las opciones del menú incluyen:


Opciones de menú	Acción de menú
NuevoClase	Abre el Asistente para clases.
NuevoInterfaz	Abre el Asistente para interfaces.
NuevoPaquete	Abre el cuadro de diálogo Crear paquete.
NuevoArchivo	Abre el cuadro de diálogo Crear archivo.
NuevoDirectorio	Abre el cuadro de diálogo Crear directorio.

- 6 Rellene los campos del asistente o del cuadro de diálogo para el archivo o paquete.

Para crear un archivo o paquete para una aplicación web (dentro del directorio o subdirectorio raíz de la aplicación web) desde el menú contextual del panel del proyecto, consulte “Directorio raíz” en la *Guía del desarrollador de aplicaciones web*. Es una función de JBuilder Enterprise.

Presentación de archivos

Para ver un archivo de un proyecto y poder examinarlo y modificarlo, haga doble clic en un nodo de archivo del panel del proyecto, o bien arrastre el archivo desde el panel del proyecto y colóquelo en el panel de contenido. 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 bien pulse el símbolo situado a la izquierda del icono o seleccione el paquete y pulse *Intro*. 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 IDE de JBuilder. Para cada archivo abierto en el proyecto, aparece una pestaña con un nombre de archivo. La barra de título de JBuilder muestra la vía de acceso completa, el nombre y la extensión del archivo. Si un archivo es de sólo lectura, su pestaña del panel de contenido muestra un icono  que indica dicho estado. Cualquier archivo de los que se presentan se puede cerrar rápidamente pulsando el icono Cerrar de la parte izquierda de su 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”, en el capítulo “Creación y gestión de proyectos” 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

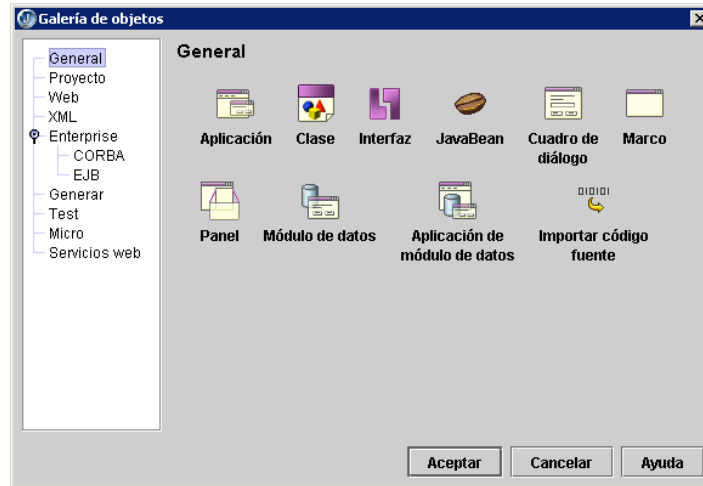
JBuilder proporciona varias formas de crear archivos Java. Los asistentes de JBuilder suelen crear y añadir varios archivos Java al proyecto. Por ejemplo, el Asistente para clases crea una clase Java, mientras que el Asistente para aplicaciones crea múltiples archivos Java que constituyen una aplicación 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 o pulse el botón Nuevo de la barra de herramientas principal. Si lo prefiere, pulse el menú desplegable del botón Nuevo y seleccione Nuevo.

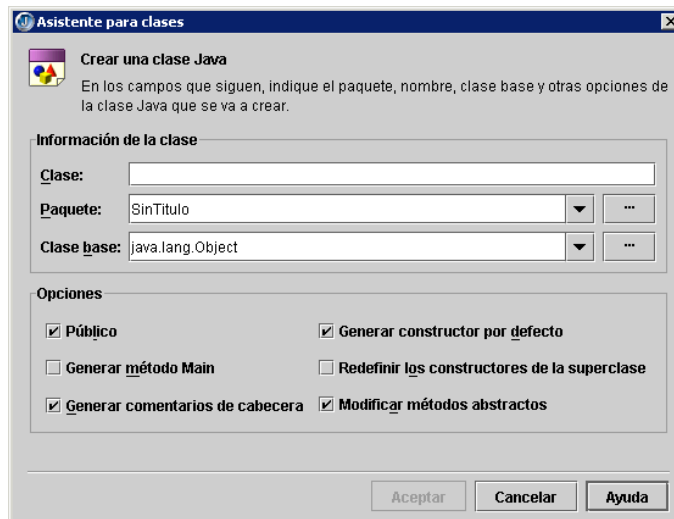
Se abre el cuadro de diálogo Galería de objetos.



- 2 Seleccione la ficha General del cuadro de diálogo Galería de objetos y haga doble clic en el icono Clase para abrir el Asistente para clases.



Si lo prefiere, pulse el menú desplegable del botón Nuevo y seleccione Clase.



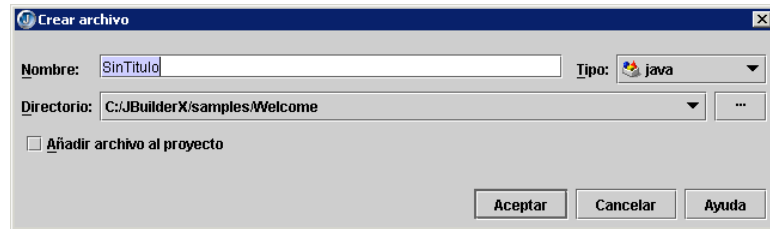
- 3 Pulse el botón Ayuda si desea más información sobre el Asistente para clases.
- 4 Rellene los campos del Asistente para clases de la forma apropiada.

5 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 haga clic con el botón derecho del ratón en un nodo de archivo, paquete o proyecto en el panel del proyecto y seleccione Nuevo|Archivo. Se abre el cuadro de diálogo Crear 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. (Se puede elegir entre muchas extensiones de archivo.)
- 4** Desplácese al directorio en el que desea guardar el archivo.
- 5** Si lo desea, active la casilla de selección Añadir archivo al proyecto.
- 6** Pulse Aceptar.
El nuevo archivo vacío aparece en el panel de contenido y su nombre aparece en el panel del proyecto.
- 7** Añada el código que desee al nuevo archivo.
- 8** Para guardar el archivo, seleccione Archivo|Guardar <"nombre de archivo"> o pulse *Ctrl+S*.

Si desea añadir el archivo al proyecto pero no activó la casilla de selección Añadir archivo al proyecto en el cuadro de diálogo Crear archivo:

- 1** Pulse el botón Añadir archivos/paquetes/clases de la barra de herramientas del panel del proyecto o haga clic con el botón derecho del ratón en el nodo del archivo de proyecto, en el panel del proyecto.
- 2** Seleccione Añadir archivos/paquetes/clases en el menú contextual.
Se abre el cuadro de diálogo Añadir archivos o paquetes al proyecto.
- 3** Abra la ficha Explorador para añadir un archivo.
- 4** Desplácese hasta el archivo recién creado, selecciónelo y pulse Aceptar.

Eliminación y borrado de archivos

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, pulse el botón Eliminar del proyecto en la barra de herramientas del panel, *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 Eliminar.

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 de proyecto.
- 2 Seleccione Proyecto|Renombrar o Archivo|Renombrar, o haga clic con el botón derecho en el panel de proyecto y seleccione Renombrar.
- 3 Escriba el nuevo nombre en el cuadro de diálogo Cambiar nombre 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”> en el menú contextual.
- 3 Escriba el nuevo nombre en el cuadro de diálogo Cambiar nombre 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.

Precaución Cuando se cambia el nombre de un proyecto o un archivo no se modifican los nombres de paquete y archivo a los que se hace referencia en el código. JBuilder Developer y Enterprise ofrecen la función de perfeccionamiento por cambio de nombre, Esto sustituye el nombre antiguo por el nuevo en todos los lugares donde aparece.

Apertura de proyectos

Para abrir un proyecto:

- 1 Seleccione ArchivosAbrir 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.

Sugerencia Es posible que el proyecto se encuentre en la lista de reapertura. Seleccione ArchivosAbrir de nuevo en el menú principal o pulse el botón Abrir de nuevo de la barra de herramientas principal para presentar la lista de proyectos.



Cierre de proyectos

Para cerrar un proyecto, siga uno de estos tres pasos:



- Pulse el botón Cerrar proyecto de la barra de herramientas del panel del proyecto.
- Pulse con el botón derecho del ratón en el archivo del proyecto (.jpx) del panel del proyecto y, a continuación, seleccione Cerrar proyecto.
- Seleccione ArchivosCerrar proyectos.

Sugerencia JBuilder añade los archivos cerrados a la lista de reapertura. Se puede acceder a esta lista desde el menú principal (ArchivosAbrir de nuevo) y desde la barra de herramientas principal (botón Abrir de nuevo archivo).



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

Más información sobre los proyectos de JBuilder

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.

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 distribuida desde la línea de comandos.
- Comprobar el applet.

Si quiere ver el código fuente asociado a este tutorial, consulte [“Código fuente del applet” en la página 8-35](#).

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 los [“Aspectos generales” en la página 8-2](#), donde se tratan importantes cuestiones sobre las applets.
- El apartado Accesibilidad en las Sugerencias de JBuilder contiene sugerencias sobre la utilización de las funciones de JBuilder para mejorar la facilidad de uso de JBuilder por parte de 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

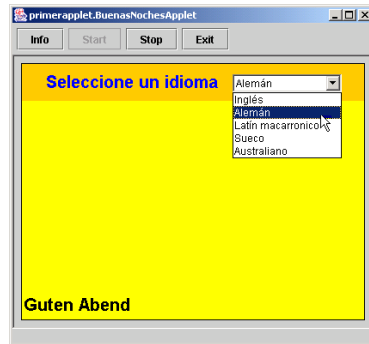
Es importante que al diseñar applets recuerde que la compatibilidad del navegador con Java puede ser limitada, según el navegador que utilice. En este tutorial se diseña el applet mediante los componentes JDK 1,4 AWT, que son compatibles con la mayoría de los navegadores, como Netscape Navigator y Microsoft Internet Explorer. No obstante, si desea diseñar un applet mediante componentes Swing, es necesario que determine las funciones Java que son compatibles con su navegador. La información de compatibilidad con el navegador para applets Java se encuentra normalmente en los contenidos de los menús Herramientas, Preferencias o Ayuda de Netscape o Internet Explorer. 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.

- Importante** Si desea obtener más información acerca de la ejecución de applets en JBuilder, consulte “Ejecución de applets” en “Las applets” de la *Guía del desarrollador de aplicaciones web*.

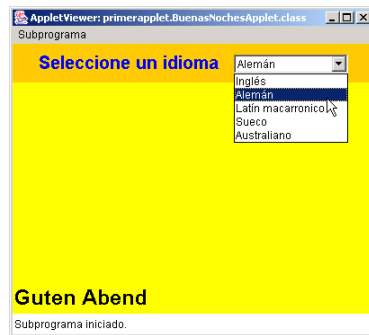
El applet Buenas noches

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:



Si ejecuta el applet ya terminada en el visor de applets de Sun, tendrá la siguiente apariencia:



Para ver el código fuente del applet completo, consulte [“Código fuente del applet” en la página 8-35](#).

Si desea obtener información más detallada acerca de las applets y su distribución, consulte “Las 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

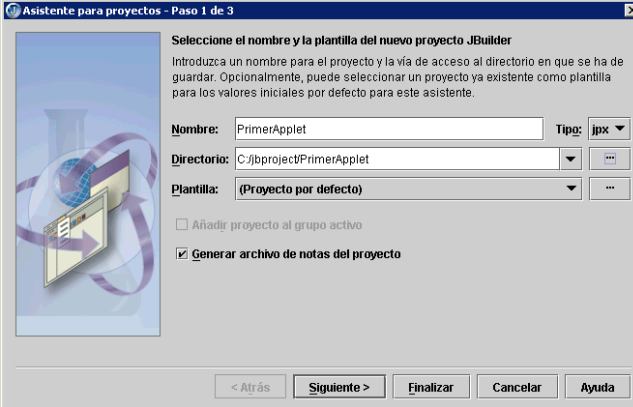
Antes de comenzar este tutorial, lea los [“Aspectos generales” en la página 8-2](#), donde se tratan cuestiones importantes sobre las applets como la compatibilidad con navegadores, las versiones del JDK y los componentes de las applets.

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

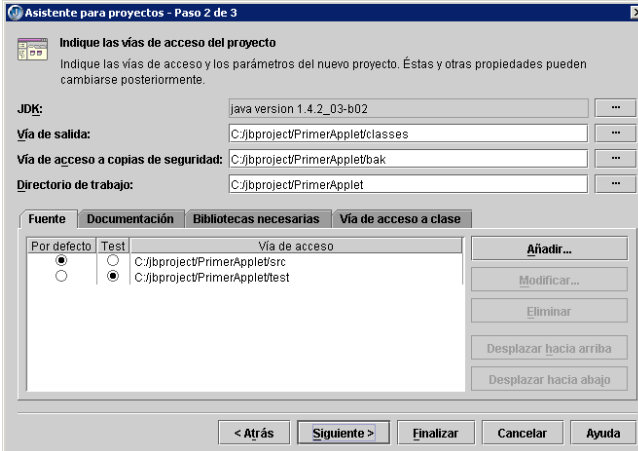
- 1 Elija Archivo|Nuevo proyecto para iniciar el Asistente para proyectos.
- 2 En el primer paso del Asistente para proyectos, realice los siguientes cambios en los campos correspondientes:

Paso 1: Creación de un proyecto

- Nota** Escriba `PrimerApplet` en el campo Nombre.
- Quando 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 `<raíz>/jbproject`. Si desea más información acerca del directorio raíz, consulte [“Convenciones de la documentación” en la página 2-4](#). Para obtener más información sobre los proyectos, consulte el [Capítulo 7, “Proyectos de JBuilder”](#).
- a** Seleccione `.jpx` en el menú desplegable Tipo.
 - c** Seleccione la opción Generar archivo de notas del proyecto.
- Quando 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 Foundation 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

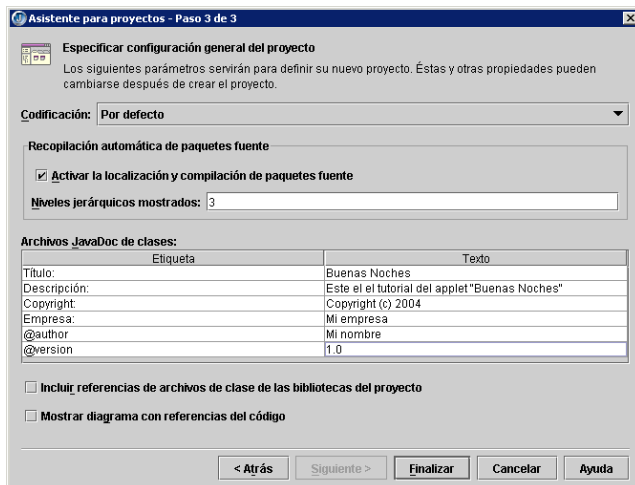
- “Configuración de las propiedades de proyecto” en *Creación de aplicaciones con JBuilder* si desea obtener más información sobre el cambio del JDK y su modificación.
- “Ejecución de applets” en “Las applets” en *Guía del desarrollador de aplicaciones web* si desea más información sobre la ejecución 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.
 - 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.

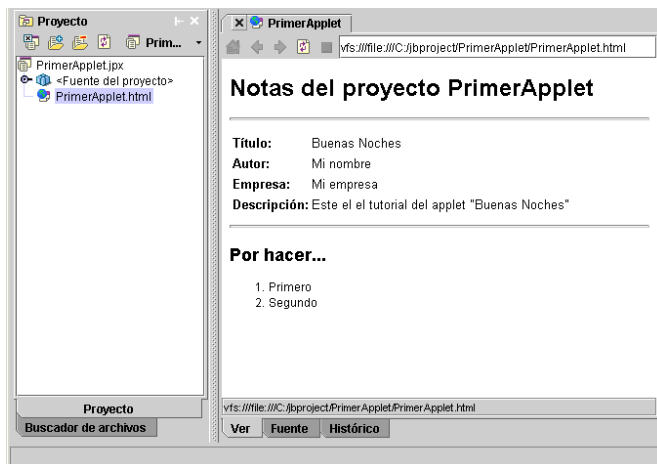
Paso 1: Creación de un proyecto

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 IDE de JBuilder.
- 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 `BuenasNochesApplet` en el campo Clase.

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

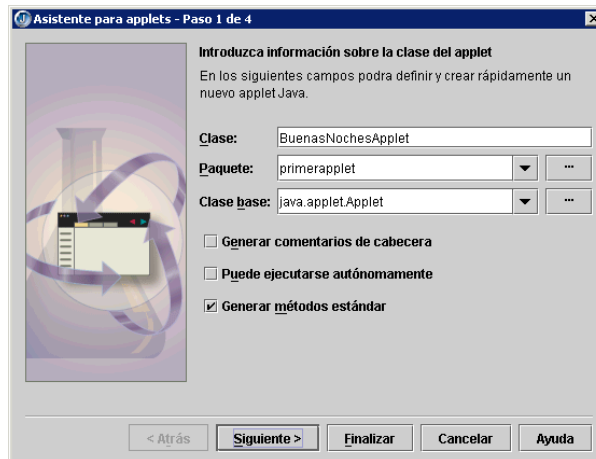
Nota El nombre completo de la clase (nombre del paquete + nombre de la clase) es `primerapplet.BuenasNochesApplet.class`.

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

Precaución Si crea el applet con `javax.swing.JApplet`, no se ejecutará en la mayoría de los navegadores. En el momento de la redacción de este manual, los navegadores no son compatibles con Swing. Para más información consulte [“Aspectos generales” en la página 8-2](#).

- 6 Active la opción Generar métodos estándar:

El Paso 1 del Asistente para applets 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, e inserta en el código fuente líneas que gestionan los parámetros. Los parámetros de las applets, equivalentes a los argumentos de línea de comandos de las aplicaciones, permiten personalizar el applet.

Nombre*	Tipo*	Descripción	Variable*	Valor por defecto
---------	-------	-------------	-----------	-------------------

Consulte

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

- 8 No añada ningún parámetro a este tutorial. Pulse Siguiete 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 Introduzca `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 applets 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 <code>codebase</code> 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 puede ver en el archivo HTML al generarlo. En este ejemplo, el nombre completo de la clase es <code>primerapplet.BuenasNochesApplet.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 recopilatorios deben estar en el directorio especificado por <code>codebase</code> .
name	Este atributo optativo indica el nombre del applet.
width/height	Estos atributos necesarios determinan la anchura y la altura en píxeles del área de representación del applet.
hspace/vspace	Estos atributos optativos determinan, en píxeles, los rellenos horizontal (márgenes izquierdo y derecho) y vertical (márgenes superior e inferior) que rodean al applet.
alinear	Este atributo optativo determina la alineación del applet en la página HTML.

Paso 2: Generación de los archivos de código fuente

Importante

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

Consulte

- “El tema de ayuda “atributos de la etiqueta < applet>” en “Las applets”, en la *Guía del desarrollador de aplicaciones web*.

10 Pulse **Siguiente** para avanzar al Paso 4 y pulse **Finalizar** para crear automáticamente una configuración de ejecución, y cierre el Asistente para applets.

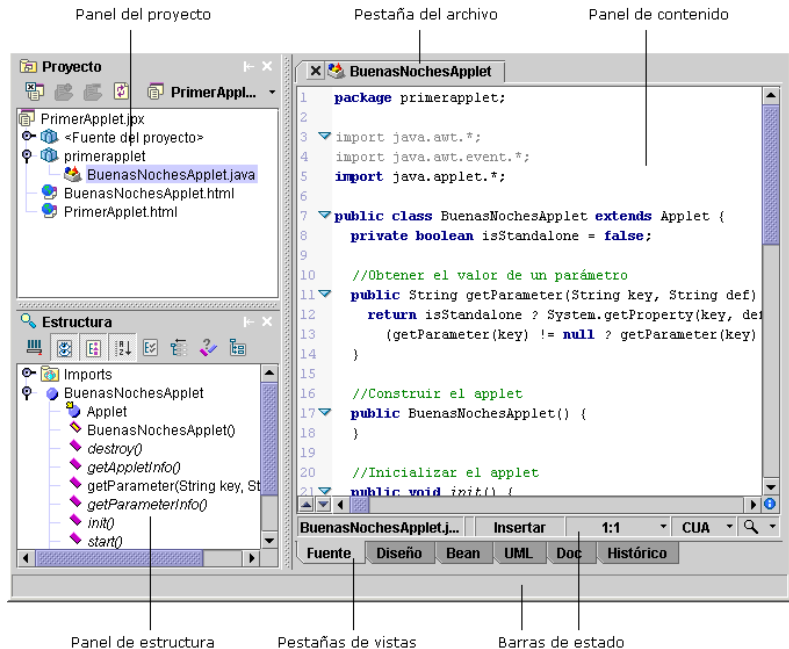


Se crean dos archivos, `BuenasNochesApplet.java` y `BuenasNochesApplet.html`, y se añaden al proyecto. `BuenasNochesApplet.java` se abre en el panel de contenido.

Nota

En el panel del proyecto también aparece automáticamente un nodo de paquete de código fuente llamado `primerapplet` 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**).

El IDE de JBuilder muestra el proyecto, los archivos, la estructura y el código fuente del proyecto del primer applet en los paneles del proyecto, de estructura y de contenido.



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

JBuilder guarda los archivos fuente en:

```
<JBuilder_projects>/FirstApplet/src/firstapplet/
```

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

```
<JBuilder_projects>/FirstApplet/classes/
```

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

```
<JBuilder_projects>/FirstApplet/classes/firstapplet/
```

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 generación incluye las tareas de generación, como la preparación de archivos que no son de Java para la compilación, la compilación de los archivos fuente Java, la copia de recursos, la creación de recopilatorios y la distribución, entre otras. 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 applets de JDK 1.1.x y 1.2/1.3 en JBuilder, consulte “Ejecución de applets” en el capítulo “Las applets” de la *Guía del desarrollador de aplicaciones web*.



Sugerencia

1 Elija Ejecutar/Ejecutar Proyecto, o pulse el botón Ejecutar para compilar y ejecutar el applet. El menú Ejecutar y el botón Ejecutar ejecutan la aplicación en AppletTestbed, el visualizador de applets de JBuilder.

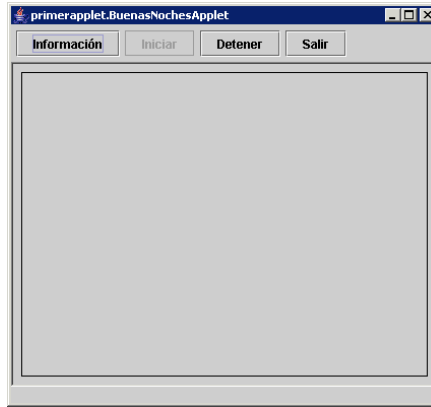
Si lo prefiere, haga clic con el botón derecho del ratón en

`BuenasNochesApplet.html`, en el panel del proyecto, y seleccione 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 IDE 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:



Los parámetros de ejecución del applet se pueden modificar por medio de la pestaña Applet de la ficha Ejecutar del cuadro de diálogo Propiedades de ejecución. Para acceder rápidamente a este cuadro de diálogo, elija Ejecutar|Configuraciones, seleccione `BuenasNochesApplet` en la ficha Ejecutar 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()`, y 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: main
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

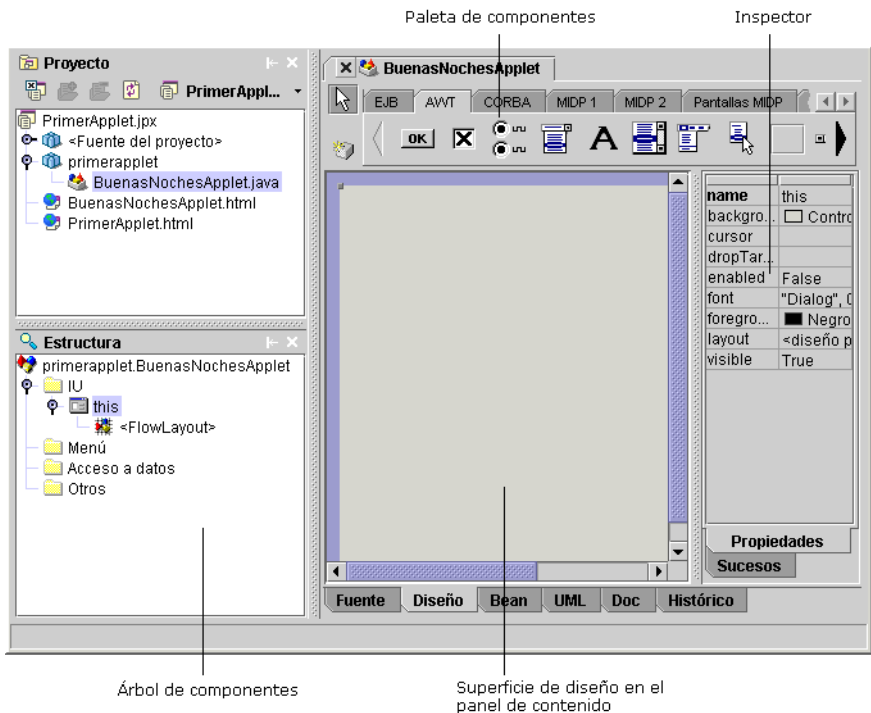
Ahora que el Asistente para applets ha generado el shell del applet, se va a personalizar con varios componentes:

- 1 Haga clic en la pestaña Diseño, en la parte inferior de `BuenasNochesClass.java`, que debe seguir abierto en el panel de contenido. De esta forma, el proyecto cambia a la vista de diseño y muestra los diversos elementos del diseñador de interfaces.

El diseñador de interfaces consta de una superficie de dibujo en el panel de contenido, con una paleta de componentes encima, en la que se pueden elegir componentes para añadirlos al diseño. Ahora, el panel de estructura muestra un árbol de componentes que presenta los elementos de la clase que se está diseñando y contiene carpetas como `IU`, `Menú`, `Acceso a datos` y `Por defecto`. El inspector se encuentra a la derecha de la superficie de diseño y sirve para configurar las propiedades de los componentes y añadir sucesos al código.

Nota El espacio de trabajo de JBuilder (esto es, la posición de los paneles en el visualizador) es personalizable, por lo que es posible que no coincida exactamente con la disposición que se muestra a continuación.

Figura 8.1 Diseñador de interfaces de usuario



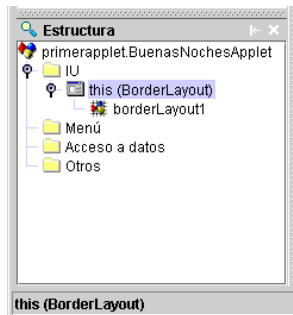
- 2 Cambie el diseño del panel principal del interfaz de usuario a `BorderLayout` en el Inspector:
 - a Seleccione el panel principal de la interfaz pulsando el nodo `this` del árbol de componentes.
 - b Haga clic a la derecha de la propiedad `layout` de la ficha Propiedades del Inspector.
 - c Seleccione `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.

Precaución

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.*`. La 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 diseño aparece como un elemento del árbol, justo debajo del supercontenedor:

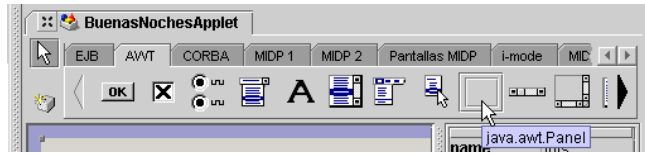


- 3 Pulse la ficha AWT de la paleta de componentes, en la parte superior del diseñador de interfaces.

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.

Precaución

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. Cerciórese de que coloca los componentes desde la ficha AWT de la paleta.



- 4 Añada dos paneles al panel `this` de la forma que se explica a continuación. 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 aparece 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. `panel1` y `panel2` se añaden bajo `this` en el árbol de componentes.



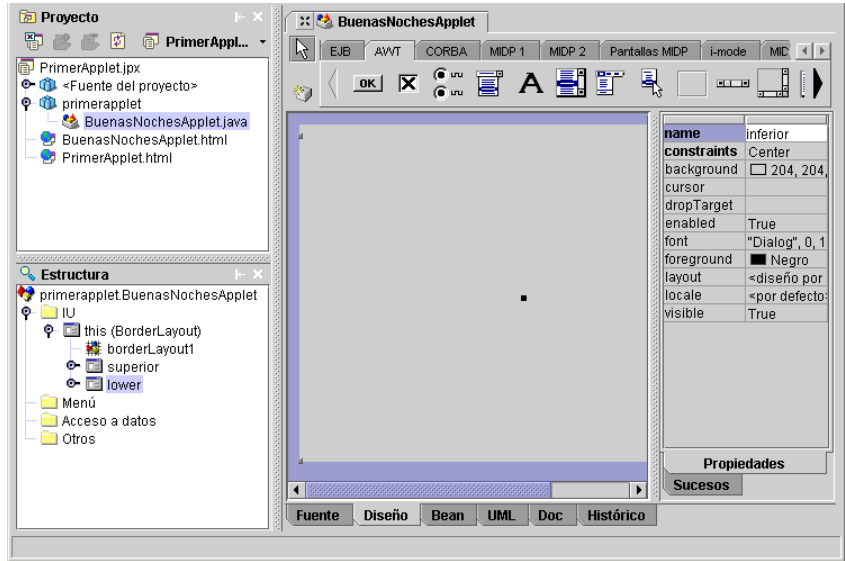
- c Haga clic en la Herramienta de selección a la izquierda de la paleta de componentes, para desactivar la selección del componente.

- 5 Modifique la propiedad `constraints` de los nuevos paneles para controlar su ubicación, como sigue:

- a 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.) Si no es así, haga clic en la columna situada a la derecha de la propiedad `constraints` y seleccione `Center` en la lista desplegable.

- b Repita el proceso con `panel2` y asígnele el valor `Center`.

- 6 Seleccione de nuevo `panel1` y, en el inspector, seleccione la propiedad `name`. Resalte el valor `panel1` en la columna derecha y escriba `upper` para sustituirlo.

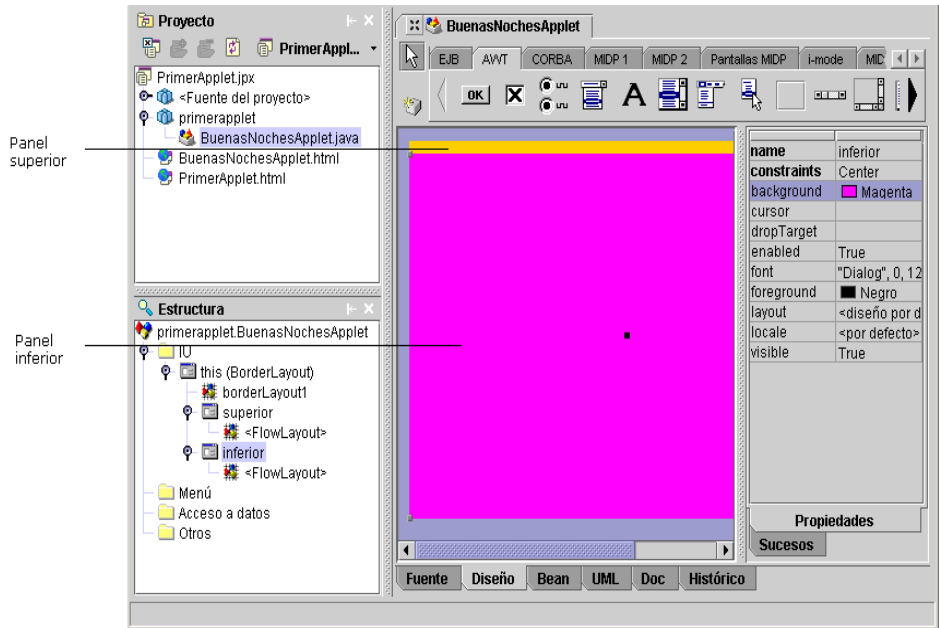


Sugerencia

Si lo prefiere, haga clic en un componente con el botón derecho del ratón, en el árbol de componentes, y elija **Cambiar nombre** en el menú.

- 7 Asigne al panel inferior el nombre `inferior`:
- 8 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.

- 9 Cambie el color de fondo de `inferior` a Magenta. Ahora la interfaz de usuario debe tener este aspecto:



- 10 Seleccione el panel `inferior` y cambie su diseño a `CardLayout` en el inspector.

El panel de `CardLayout` contendrá 5 paneles, cada uno de los cuales mostrará el texto “Buenas noches” en cinco idiomas distintos. `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.

- 11 Añada cinco paneles (`panel1` a `panel5`) al panel `inferior` pulsando **Mayús+Click** mientras selecciona el componente `Panel`. Para asegurarse de que todos ellos se añaden al panel correcto, pulse cinco veces el nodo `inferior` del panel en el árbol de componentes. Cada uno de estos paneles contendrá el texto “Buenas noches” en un idioma distinto. Elija la herramienta de selección para desactivar la selección múltiple después de soltar `jpanel5`.

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

- 12 Cambie el diseño de `panel1` a `panel5` a `BorderLayout`. Esto se puede hacer con todos los paneles a la vez:

Pulse *Mayús+Clic* o *Ctrl+Clic* para seleccionar los cinco paneles en el árbol de componentes y cambie la propiedad `layout` a `BorderLayout` en el inspector.

13 Seleccione otro componente del árbol de componentes o del diseñador de interfaces de usuario para que los paneles dejen de estar seleccionados.

14 Asigne a cada uno de los cinco paneles un color distinto en el Inspector. La selección de colores depende del usuario.

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.

15 Guarde el archivo y el proyecto.

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

Quando 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.

17 Salga del applet.

18 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`.

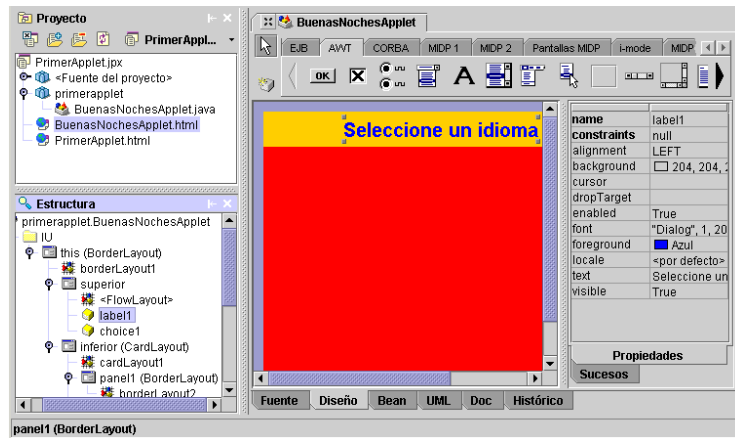
Observe que se ha añadido `label1` a `superior` en el árbol de componentes.

4 Seleccione `label1` en el árbol de componentes y haga lo siguiente:

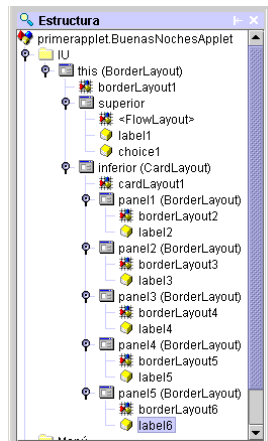
Paso 5: Incorporación de componentes AWT al applet

- a Haga doble clic en la columna de la derecha de la propiedad `text` del Inspector para resaltar el texto actual.
- b Introduzca *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 Marque **Negrita**. Intro **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 ofrece un aspecto similar a éste (con excepción de los colores):



- 5 Pulse *Mayús+Clic* para seleccionar el componente de etiqueta AWT en la paleta de componentes y vaya haciendo clic en los nodos de los paneles (`panel1` a `panel5`) para añadirles etiquetas.





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 múltiple. Si añade demasiados paneles por error, seleccione los sobrantes en el árbol de componentes y elimínelos

- 6 Escriba “Buenas noches” en un idioma distinto en cada etiqueta. Primero, seleccione la etiqueta que hay debajo de cada panel 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: ¡Buenas noches! (Español)`
 - `label3: "Guten Abend." (Alemán)`
 - `label4: Buona sera! (Italiano)`
 - `label5: Bonsoir ! (Francés)`
 - `label6: Goede avond! (Neerlandés)`
- 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. Deje la propiedad `foreground` de las etiquetas (el color del texto) como **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: Centro`

Nota

Observe la posición de cada etiqueta (como indican los tiradores de selección) en el panel `BorderLayout`. **Center** ocupa todo el panel, aunque el texto queda a la izquierda a causa de la alineación izquierda, mientras que **North**, **South**, **East** y **West** sólo ocupan una parte.

- 9 Ahora, seleccione `panel1` a `panel5` y cambie su diseño a `FlowLayout`, que en realidad es el diseño por defecto del componente `Panel` AWT.

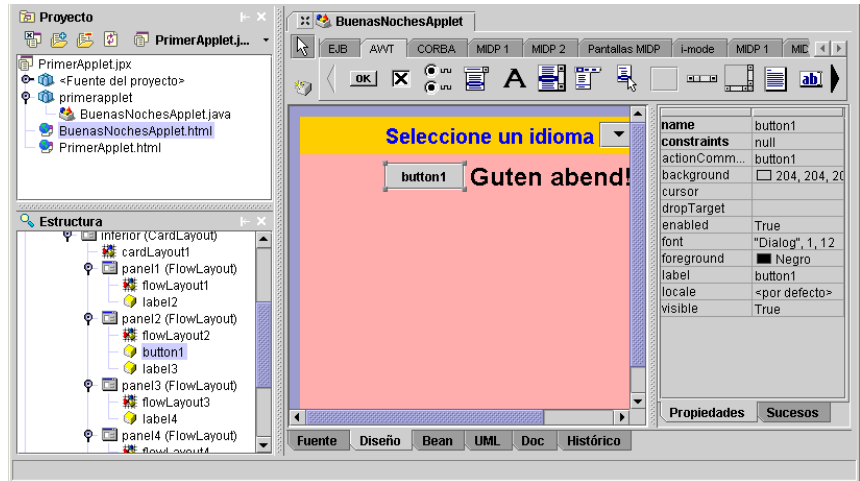
Observe que la etiqueta se desplaza hasta la parte central superior de los paneles. `FlowLayout` es una buena elección de diseño si desea distribuir los componentes en filas.

A continuación se añade un botón a uno de estos paneles, 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. Cuando se pulse este botón, cambiará el color del texto de `label` del panel.



10 Haga clic en el componente Button de AWT, en la paleta, y suéltelo en panel2, a la izquierda de la etiqueta “Guten abend!”.

El panel presenta un aspecto similar al siguiente:



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, antes o después de ésta.

11 Seleccione Archivo|Guardar todo para guardar el proyecto.

Paso 6: Modificación del código fuente

En este paso se añaden los idiomas a la lista desplegable y después se añaden sucesos que asocian los paneles de idiomas al 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

Para buscar un método en el panel de estructura haga clic en el panel y escriba 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 panel de proyecto y de estructura, seleccione **Vista|Mostrar todo**.

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

```
//Inicializar el applet
public void init() {
    choice1.addItem("Inglés");
    choice1.addItem("Alemán");
    choice1.addItem("Italiano");
    choice1.addItem("Francés");
    choice1.addItem("Holandés");

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

Sugerencia

Utilice CodeInsight para completar el código. Intro `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*. CodeInsight se puede configurar en el cuadro de diálogo Preferencias (Herramientas|Preferencias|Editor|CodeInsight).

```
/**Inicializar el applet*/
```

```
public void init() {
```

```
    choice1.
```

```
    choice1.
```

```
    choice1.
```

```
    choice1.
```

```
    choice1.
```

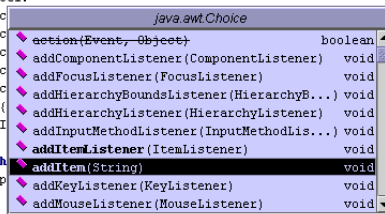
```
    try {
```

```
        jbI
```

```
    }
    catch
```

```
    {
        e.p
```

```
    }
}
```



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

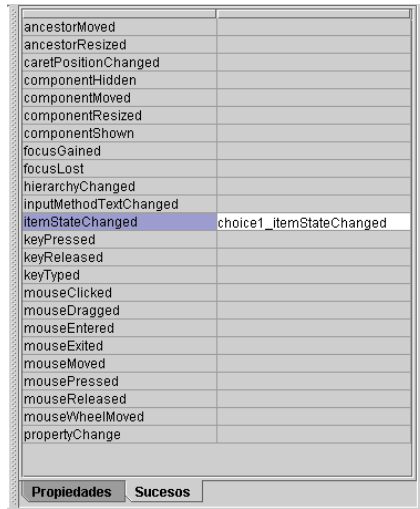
- “Acerca de los mensajes de advertencia y error” en *Creación de aplicaciones con JBuilder*.

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

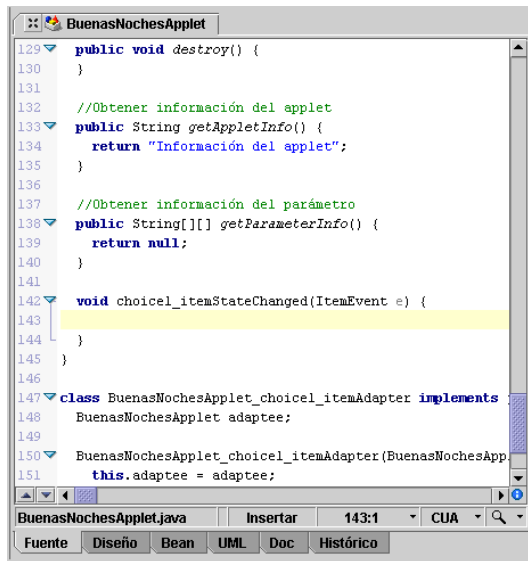
2 Asocie los sucesos de la lista `Choice` de la siguiente manera:

- Vuelva al diseñador de interfaces de usuario.
- Seleccione `choice1`, situado bajo `superior` en el árbol de componentes.
- 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.



- 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(inferior, "panel1");
    }
    else if ("Alemán".equals(choice1.getSelectedItem())){
        cardLayout1.show(inferior, "panel2");
    }
}
    
```



```

    }
    else if ("Italiano".equals(choice1.getSelectedItem())){
        cardLayout1.show(inferior, "panel3");
    }
    else if ("francés".equals(choice1.getSelectedItem())){
        cardLayout1.show(inferior, "panel4");
    }
    else if ("Holandés".equals(choice1.getSelectedItem())){
        cardLayout1.show(inferior, "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 `BuenasNochesClase.html` en el panel del proyecto y seleccione Ejecutar utilizando “BuenasNochesClase”.

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 IDE de JBuilder. 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 `botón1` en el `panel2`. Cuando se pulsa el botón, el texto “Guten abend!” de la etiqueta `label6` cambia a amarillo.

- 1 Añada una etiqueta y un suceso de botón de la forma que se explica a continuación:
 - a Vuelva a cambiar al diseñador de interfaces y seleccione `button1` en `panel2`. Cambie la propiedad del botón `label` en la ficha Propiedades del Inspector, de `button1` a `Púlsame`. Pulse *Intro*. Observe que el botón cambia de tamaño automáticamente para adaptarse al texto.
 - b Pulse la pestaña Sucesos del Inspector para definir lo que ocurre cuando se pulsa `jButton4`.
 - c 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 amarillo el texto “Guten abend!”.

- d Escriba el siguiente código, marcado en negrita:

```
void button1_actionPerformed(ActionEvent e) {  
    label13.setForeground(new Color(255,255,0));  
}
```

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

“Guten abend!” debe cambiar a amarillo.

El applet presenta 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 `codebase` especifica la ubicación correcta del archivo de clase con relación al archivo HTML. En este ejemplo, el valor de `codebase` es `."`, porque el archivo JAR que contiene el archivo de clase estará en el mismo directorio que el archivo HTML.
 - 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`.

La etiqueta applet tiene este aspecto:

```
<applet
  codebase = "."
  code = "primerapplet.BuenasNochesClase.class"
  name     = "TestApplet"
  width    = "400"
  height   = "300"
  hspace   = "0"
  vspace   = "0"
  align    = "middle"
>
</applet>
```

- 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.

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

Consulte

- “Distribución de programas en Java” en *Creación de aplicaciones con JBuilder*.
- El paso 16 del tutorial de JBuilder “Creación de un editor de texto en Java - Distribución de la aplicación Editor de texto”, en *Diseño de aplicaciones con JBuilder*.

Distribución del applet con el Creador de recopilatorios

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

Para distribuir la aplicación:

- 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 `/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:

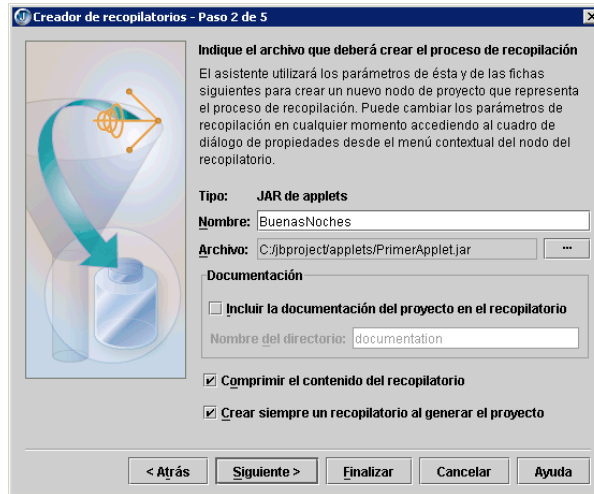


- 5 Seleccione Applet JAR en la lista desplegable Tipos de recopilatorio del Paso 1. Pulse Siguiente para avanzar al Paso 2.

Nota Seleccione el tipo de archivo Applet ZIP si desarrolla applets para navegadores antiguos incompatibles con los archivos JAR.

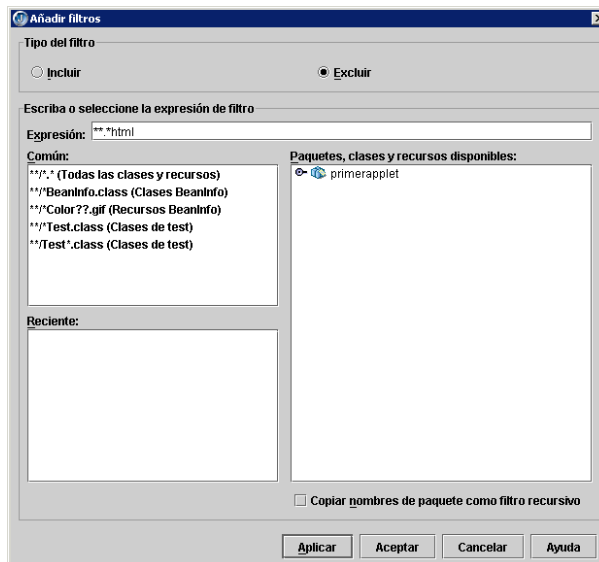
- 6 Acepte el nombre por defecto de `Applet JAR` en el campo Nombre.
- 7 Pulse el botón de puntos suspensivos contiguo al campo Archivo y desplácese al directorio `/<raíz>/jbproject/applets/`.
- 8 Cambie el nombre del archivo JAR en el campo Nombre del archivo por el de `BuenasNoches.jar` y pulse

El Paso 2 tiene el siguiente aspecto:



- 9 Pulse Siguiente para pasar al Paso 3.
- 10 Pulse Añadir filtros.

Aparece el cuadro de diálogo Añadir filtros.



11 Pulse el botón circular Excluir del grupo Tipo de archivo.

12 Escriba `**/*.html` en el campo Expresión, en el grupo Escriba o seleccione la expresión de filtro.

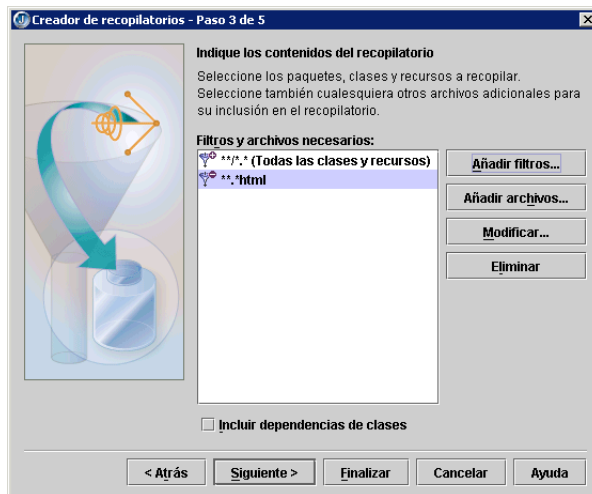
13 Pulse Aplicar.

14 Pulse Aceptar.

El Creador de recopilatorios añade la entrada `**/*.html` al cuadro de lista Clases y recursos especificados, precedida por un signo menos (-). Este filtro excluye el archivo HTML del applet, `BuenasNochesClass.html`, del archivo JAR. Más adelante se copiará el archivo `BuenasNochesClass.html` al directorio `applets`.

15 Acepte las otras opciones por defecto del Paso 3.

El Paso 3 tiene el siguiente aspecto:



16 Pulse Siguiente para pasar al Paso 4.

17 Pulse Siguiente para pasar al Paso 5.

No es necesario añadir ninguna biblioteca.

18 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 “Distribución de programas en Java”, en *Creación de aplicaciones con JBuilder*.

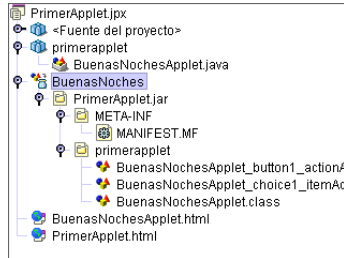
19 Pulse Finalizar para salir del Creador de recopilatorios.

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.

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

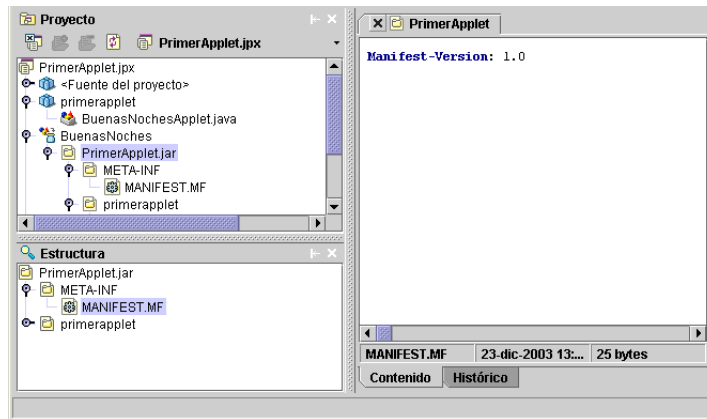
El Creador de recopilatorios recoge el paquete `primerapplet`, que se ha añadido al archivo JAR en el tercer paso del asistente.

21 Pulse el icono de ampliación situado junto al nodo del archivo Applet JAR para ver el recopilatorio `BuenasNoches.jar`.



22 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 por separado, en vez de incluirlos en todos los recopilatorios JAR.

Consulte

- “Utilización del Creador de Recopilatorios” en *Creación de aplicaciones con JBuilder*.
- “Distribución 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.BuenasNochesClase.class"
  archive = "BuenasNoches.jar"
  name     = "TestApplet"
  width    = 400
  height   = 300
  hspace   = 0
  vspace   = 0
  align    = top
>
</applet>
```

Sugerencia

Si tiene varios archivos JAR para el applet, enumérellos separados con comas, como se indica a continuación:

```
archive="archivo1.jar, archivo2.jar"
```

Importante

Algunos navegadores antiguos no son compatibles con los archivos JAR ni con las listas de 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.BuenasNochesClase.class"
```



```
archive = "BuenasNoches.jar"
name    = "TestApplet"
width   = 400
height  = 300
hspace  = 0
vspace  = 0
align   = top
```

>

Para ver este applet se necesita un navegador compatible con Java que utilice el JDK 1.1.x o posterior.

</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 `code`. Si estos valores son incorrectos, el applet no se ejecutará. Recuerde que el valor `codebase` designa la ubicación del código del applet (archivo CLASS o JAR) en relación con el 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`.

Precaución

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, `BuenasNochesClass.html`, que contiene la etiqueta <applet> y se encuentra en el directorio `src` del proyecto. Si se copia `PrimerApplet.html` en vez de `BuenasNochesClass.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 `BuenasNochesClass.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 BuenasNochesClass.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

En Windows, utilice barras invertidas (`\`).

Si el applet “Buenas noches” se carga y se ejecuta en el `appletviewer`, la distribución habrá sido correcta y todas las clases se habrán encontrado e incluido. Si el applet no se ejecuta, compruebe los mensajes de error, corríjalos, vuelva a compilar, distribuya y haga otra prueba.

Hay un ejemplo del applet “Buenas noches” en el subdirectorio `samples/Tutorials/FirstApplet/` del directorio de la instalación de JBuilder.

Si tiene problemas al ejecutar el applet, consulte “Código fuente del applet” en la [página 8-35](#). Para errores habituales, consulte:

- “Solving common applet problems” en <http://www.java.sun.com/docs/books/tutorial/applet/problems/index.html>
- “Errores comunes en la etiqueta `<applet>`” y “Sugerencias adicionales para que funcionen las applets” en el capítulo “Las applets” de 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 profundiza en la comprobación de applets, en “[Comprobación del applet distribuido en la Web](#)” en la [página 8-35](#) se encuentra una visión 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 las applets, consulte:

- “The Java(tm) Tutorial” en <http://java.sun.com/docs/books/tutorial/index.html>
- Charlie Calvert, Part II: Applets 2 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 distribuido 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 recopilatorio 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. Compruebe también si hay mensajes de error en la Consola Java del navegador.

Para abrir la Consola Java:

- Seleccione Communicator|Herramientas|Consola Java en Netscape.

La disponibilidad y el acceso a la Consola Java varían según la versión del navegador.

- 3 Corrija todos los errores, vuelva a distribuir el applet y compruébelo de nuevo en los navegadores.

Consulte

- “Comprobación de applets” en *Guía del desarrollador de aplicaciones web*

Código fuente del applet

Para ver el código fuente, consulte:

- “Código fuente HTML del applet BuenasNochesClass.html” en la [página 8-36](#)

- [“Código fuente de la clase del applet BuenasNochesClase.java” en la página 8-36](#)

Código fuente HTML del applet BuenasNochesClass.html

```
<html>
<head>
<title>
  Página de prueba HTML de Buenas noches
</title>
</head>
<body>
  primerapplet.BuenasNochesApplet aparece debajo en los navegadores que aceptan Java.<br>
<applet
  codebase = "."
  code = "primerapplet.BuenasNochesClase.class"
  archive = "BuenasNoches.jar"
  name     = "TestApplet"
  width    = "400"
  height   = "300"
  hspace   = "0"
  vspace   = "0"
  align    = "middle"
>
  Para ver este applet se necesita un navegador compatible con Java que utilice el JDK
  1.1.x o posterior.
</applet>
</body>
</html>
```

Código fuente de la clase del applet BuenasNochesClase.java

```
package primerapplet;

import java.awt.*;
import java.awt.event.*;
import java.applet.*;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JButton;
import javax.swing.UIManager;

public class BuenasNochesClass extends Applet {
  private boolean isStandalone = false;
  BorderLayout BorderLayout1 = new BorderLayout();

  Panel superior = new Panel();
  Panel inferior = new Panel();
  CardLayout cardLayout1 = new CardLayout();
  Choice choice1 = new Choice();
  Label label1 = new Label();
  Panel panel1 = new Panel();
  Panel panel2 = new Panel();
  Panel panel3 = new Panel();
  Panel panel4 = new Panel();
  Label label2 = new Label();
  Label label3 = new Label();
  Label label4 = new Label();
  Label label5 = new Label();
  Panel panel5 = new Panel();
```

```

Label label6 = new Label();
Button button1 = new Button();//Obtener el valor de un parámetro
public String getParameter(String key, String def) {
    return isStandalone ? System.getProperty(key, def) :
        (getParameter(key) != null ? getParameter(key) : def);
}

//Generar el applet
public GoodEveningClass() {
}
//Inicializar el applet
public void init() {
    choice1.addItem("Inglés");
    choice1.addItem("Alemán");
    choice1.addItem("Italiano");
    choice1.addItem("Francés");
    choice1.addItem("Holandés");

    try {
        jbInit();
    }
    catch(Exception e) {
        e.printStackTrace();
    }
}
//Inicialización del componente
private void jbInit() throws Exception {
    this.setLayout(borderLayout1);
    upper.setBackground(Color.orange);
    lower.setBackground(Color.magenta);
    inferior.setLayout(cardLayout1);
    label1.setFont(new java.awt.Font("Dialog", Font.BOLD, 20));
    label1.setForeground(Color.blue);
    label1.setText("Seleccione un idioma");
    panel1.setBackground(new Color(255, 255, 140));
    panel2.setBackground(new Color(198, 0, 198));
    panel3.setBackground(new Color(204, 204, 255));
    panel4.setBackground(new Color(255, 206, 255));
    label2.setFont(new java.awt.Font("Dialog", 1, 24));
    label2.setLocale(java.util.Locale.getDefault());
    label2.setText("Buenos días");
    label3.setFont(new java.awt.Font("Dialog", 1, 24));
    label3.setText("Guten abend! ");
    label4.setFont(new java.awt.Font("Dialog", 1, 24));
    label4.setText("Buona sera!");
    label5.setFont(new java.awt.Font("Dialog", 1, 24));
    label5.setText("Bonsoir ! ");
    panel5.setBackground(new Color(204, 255, 255));
    label6.setFont(new java.awt.Font("Dialog", 1, 24));
    label6.setText("Goede avond!");
    button1.setLabel("Púlsame");
    button1.addActionListener(new BuenasNochesClass_button1_actionAdapter(this));
    choice1.addItemListener(new BuenasNochesClass_choice1_itemAdapter(this));
    superior.add(label1);
    superior.add(choice1);
    this.add(inferior, BorderLayout.CENTER);
    this.add(superior, BorderLayout.NORTH);
    inferior.add(panel1, "panel1");
    panel1.add(label2, null);
    inferior.add(panel2, "panel2");
    panel2.add(button1, null);
    panel2.add(label3, null);
    lower.add(panel3, "panel3");
    panel3.add(label4, null);
    lower.add(panel4, "panel4");
    panel4.add(label5, null);
    lower.add(panel5, "panel5");
}

```

Código fuente del applet

```
        panel5.add(label6, 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 "Información sobre el applet";
    }
    //Obtener información del parámetro
    public String[][] getParameterInfo() {
    return null;
    }

    void choice1_itemStateChanged(ItemEvent e) {
        if ("Español".equals(choice1.getSelectedItem())){
        cardLayout1.show(inferior, "panel1");
        }
        else if ("Alemán".equals(choice1.getSelectedItem())){
        cardLayout1.show(inferior, "panel2");
        }
        else if ("Italiano".equals(choice1.getSelectedItem())){
        cardLayout1.show(inferior, "panel3");
        }
        else if ("francés".equals(choice1.getSelectedItem())){
        cardLayout1.show(inferior, "panel4");
        }
        else if ("Holandés".equals(choice1.getSelectedItem())){
        cardLayout1.show(inferior, "panel5");
        }
    }

    void button1_actionPerformed(ActionEvent e) {
    label3.setForeground(new Color(255,255,0));
    }
}

class BuenasNochesClass_choice1_itemAdapter implements java.awt.event.ItemListener {
    BuenasNochesClass adaptee;

    BuenasNochesClass_choice1_itemAdapter(BuenasNochesClass adaptee) {
    this.adaptee = adaptee;
    }
    public void itemStateChanged(ItemEvent e) {
        adaptee.choice1_itemStateChanged(e);
    }
}

class BuenasNochesClass_button1_actionAdapter implements java.awt.event.ActionListener {
    BuenasNochesClass adaptee;

    BuenasNochesClass_button1_actionAdapter(BuenasNochesClass adaptee) {
    this.adaptee = adaptee;
    }
    public void actionPerformed(ActionEvent e) {
        adaptee.button1_actionPerformed(e);
    }
}
```

Índice

A

- accesibilidad
 - desplazarse por la ayuda 2-7
- acceso, teclado 2-16
- advertencia de sintaxis no válida 6-7
- ámbito del código, ScopeInsight 5-8
- ampliar el texto del código fuente 7-2
- añadir
 - aplicaciones a proyectos 4-13
 - archivos a proyectos 4-13
 - clases a proyectos 4-13
 - macros de plantilla 6-19
 - macros, plantilla de código 6-19
 - paquetes a proyectos 4-13
 - sentencias de importación 6-19
- aplicaciones
 - tutorial 4-2
- aplicar filtros a paquetes 4-11
- applets
 - tutorial 4-27
- árboles 5-1
 - buscar 4-28
 - desplazamiento 5-1
- archivos 7-4
 - abrir en nuevos visualizadores de JBuilder 4-2
 - añadir al proyecto 7-4
 - borrar 7-10
 - cambiar nombre 7-2
 - crear vacío 7-7
 - de proyecto, ver estructura 4-17
 - eliminar 5-4
 - ver archivos de proyecto 7-7
 - ver estructura 4-17
 - ver varias 4-10
- asistencia a desarrolladores 2-17
- asistentes 6-4
 - Creador de compiladores 6-4
 - galería de objetos 6-4
 - para proyectos 7-10
- atajos de codificación 6-14
 - ErrorInsight 6-14
 - JavadocInsight 6-17
- aumentar el tamaño del texto del código 5-9
- ayuda
 - inmediata 4-30
 - online 2-6

B

- barra de división, visualizador 4-20

- barras
 - de estado, visualizador 4-27
 - de herramientas, ventana principal 4-4
- bloques de código, englobar entre llaves 6-19
- Borland
 - asistencia
 - a desarrolladores 2-17
 - técnica 2-17
 - contacto 2-17
 - correo electrónico 2-19
 - grupos de noticias 2-18
 - informar sobre errores 2-19
 - recursos en línea 2-18
 - World Wide Web 2-18
- buscar 4-28, 5-15
 - (menú) 5-13
 - buscar método redefinido 5-15
 - buscar superclase 5-15
 - clases 4-29
 - con Buscar referencias 5-15
 - con escritura incremental 4-28
 - definiciones de símbolos 5-14
 - en árboles 4-28
 - en el editor 5-15
 - en el panel del proyecto 6-4
 - en el visualizador 4-1
 - método redefinido 5-15
 - referencias 5-15
 - superclase 5-15
 - texto 5-13
 - texto en Ayuda
 - búsqueda completa de texto 2-9
 - tema actual 2-10
 - utilizar Buscar definición 5-14

C

- campos, crear 6-5
- Carpeta Por hacer 4-19
- clases 4-29
 - base, crear 6-5
 - buscar 4-29
 - internas, crear 6-5
- ClassInsight 6-9
- CodeInsight 6-7
 - configuración de teclado 6-8
 - configurar 6-11
 - fallos 6-8
- codificación 2-11, 6-1
- código
 - abreviados 6-6

- aplicar formato 5-13
- automatizar 6-1
- CodeInsight 6-7
- completar automáticamente 6-9
- configuración de teclado 6-8
- creación en el editor 2-11
- dar formato 4-26
- profundizar 4-26
- comandos
 - de búsqueda 5-13
- comandos de método abreviado
 - codificación 6-6
 - para Javadoc 2-15
- comandos de método abreviado 6-6
- comentarios, accesos directos de Javadoc 5-16
- configuración
 - del espacio de trabajo 4-6
 - desactivar 4-7
 - dividir pestañas 4-10
 - división del panel de contenido 4-10
 - gestionar 4-8
 - iconizar, integrar y segregar paneles 4-9
 - maximizar y restablecer paneles 4-8
 - restablecer original 4-7
 - ubicaciones de menú 4-8
 - del teclado
 - asignar nuevas teclas abreviadas 5-4
 - copiar 5-5
 - emulaciones de editores 5-3
 - impresión 5-5
- configuración del teclado 5-7
 - editar 6-13
- configurar 4-6
 - JBuilder, CodeInsight 6-11
 - paneles del IDE 4-6
- conflictos, resolver Javadoc 6-19
- constructores, crear 2-16
- convenciones de la documentación 4-25
 - convenciones de plataformas 2-11
- Copia de texto, el panel de mensajes 2-17
- copiar
 - del visualizador de la ayuda 5-11
 - referencias de configuraciones de teclado 5-5
- crear 2-19
- Cuadro de diálogo Opciones 6-18
- Cumplimiento de la sección 508 5-10
- Cumplimiento de los requisitos del Gobierno estadounidense. 5-10

D

- dar formato 4-26, 5-13
 - aplicar al código 4-26, 5-13
 - aplicar llaves a los bloques de código 4-27

- depurador 4-27
- depurar 4-19, 4-30
- desplazamiento
 - comandos abreviados de teclado 2-13
 - con el teclado 2-16
 - paneles del visualizador 6-11
 - por el código 4-2
 - ocultar código 4-2
 - ScopeInsight 2-1
 - sin el ratón 2-16
 - Visualizador de la ayuda 6-2
- diseñadores 5-16
 - de interfaces de usuario 4-1
 - diseñador de columnas 5-16
 - Diseñador de interfaces de usuario 5-16
 - diseñador de menús 5-16
- documentación 2-15
 - de Javadoc
 - añadir comentarios 5-16
 - comandos de método abreviado 2-15
 - de referencia 6-23
 - diseño 6-23
 - de referencia de las bibliotecas 6-23
 - de referencia de las clases 6-23
 - JBuilder 2-15

E

- editar
 - plantillas de código 6-21
 - Sync Edit 5-14
- editor
 - añadir marcadores 4-3
 - buscar 5-13
 - buscar definición de símbolo 5-15
 - buscar definiciones 5-15
 - buscar elementos de código 5-15
 - buscar método redefinido 5-15
 - buscar métodos que redefinen a otros 5-3
 - buscar superclase 5-15
 - buscar variables sin utilizar 5-3
 - búsqueda de referencias a símbolos 5-18
 - configuración del teclado 5-7
 - dividir la vista 6-10
 - etiquetas @todo 6-11
 - imprimir código fuente 5-19
 - JavadocInsight 6-17
 - mostrar números de línea 5-8
 - opciones de color 7-11
 - plantillas de código 6-5
 - seleccionar texto 8-1
 - sugerencias de Javadoc 2-15
 - trabajo con texto 5-12
 - utilizar 2-11

- ejecutar aplicaciones 4-12, 4-16
- elementos
 - de código, crear 6-14
 - de pantalla, personalizar 4-21
- emulaciones de editores 5-7
- enlaces rápidos, favoritos 5-9
- errores 5-19
 - de código 6-14
 - ErrorInsight 6-14
 - de sintaxis, ver y buscar 5-20
 - ErrorInsight 5-19
 - soluciones 5-19
- ErrorInsight 5-19
- etiquetas @todo 6-11
- presentación 4-26
- examinar clases 4-29
- expresiones 6-11
 - evaluación automática 4-30
 - ExpressionInsight 6-11
- ExpressionInsight 4-19

F

- favoritos, enlaces rápidos 5-9
- fichas
 - Depurar (cuadro de diálogo Propiedades de proyecto) 4-27
 - Punto de Observación 4-27
- filtrar, panel de estructura 4-17
- filtros del panel de estructura 6-3
- flotante (panel de mensajes) 4-11
- fuentes 4-25
 - cambiar de tamaño en el editor 7-2
 - cambiar de tamaño en el IDE 7-2
 - Convenciones empleadas en la documentación de JBuilder 4-25
- funciones de texto del editor 5-12

G

- galería de objetos 7-4
 - asistentes 6-4
- generar proyectos 2-19
 - añadir archivos 6-1
- grupos de noticias 2-18
 - Borland y JBuilder 2-18
 - public 6-4
 - Usenet 6-4

H

- herramientas 6-5
 - asistentes 6-2, 6-5
 - crear elementos de código 6-3
 - de diseño visual 4-1

- diseño visual 4-1, 6-5
- galería de objetos 4-17, 6-5

I

IDE

- barra de herramientas 4-4
- configuración del espacio de trabajo 4-6
- depurador 4-27
- JBuilder 4-32
- menús 6-9
- panel de contenido 4-23
- panel de estructura 6-11
- panel de mensajes 4-26
- panel del proyecto 6-4
- personalizar 5-15
- identificadores, buscar definición 2-17
- identificar
 - métodos que redefinen a otros 6-17
 - editor 5-23
 - ScopeInsight 5-14
 - variables sin utilizar 6-17
 - editor 5-23
- importaciones automáticas para plantillas 5-16
- plantillas de código Java 5-16
- impresión 5-19
 - código fuente 5-19
 - configuración del teclado 5-5
- informar sobre errores 2-19
- insertar clases, ClassInsight 6-9
- Ir a la línea 5-8

J

- JAR ejecutable 4-12, 4-16
- Javadoc 5-13
 - Carpeta Por hacer 4-19
 - crear 6-5
 - etiquetas @todo 6-11
 - resolver conflictos Javadoc 6-19
- JBuilder
 - grupos de noticias 2-18
 - IDE 4-32
 - informar sobre errores 2-19
 - menús 6-9
 - paseo de introducción 4-27
 - visualizador 4-32

L

- libros, JBuilder 2-15
- listas de parámetros 6-17
- llaves, englobar 6-19
- lupa (herramienta) 3-1

M

- macros 5-20
 - añadir plantillas de código 2-11
 - de plantilla 5-20
- manuales, JBuilder 2-15
- marcadores
 - editor 4-3
 - Visor de ayuda 4-3
- margen
 - números de línea 5-8
 - utilizar para seleccionar texto 5-15
- MemberInsight 6-9
- mensajes de error
 - ver y buscar 4-20
- menús 4-3
 - atajos de barra de herramientas 4-4
 - comandos 6-4
 - IDE 6-9
 - JBuilder 6-9
 - principal 4-3
- métodos abreviados de codificación
 - asistentes 6-7
 - ClassInsight 6-14
 - CodeInsight 6-9
 - crear elementos de código 7-11
 - ErrorInsight 6-13
 - galería de objetos 7-4
 - MemberInsight 6-10
 - ocultar código 6-17
 - ParameterInsight 6-11
 - plantillas de código 6-23
 - ScopeInsight 6-25
 - Sync Edit 6-5
 - TagInsight 5-4
- métodos, crear 2-16
- miembros de clase, crear 6-14
- modificar configuraciones de teclado 6-13
- modo de asociación de teclas 3-1
- mostrar el panel de mensajes 4-26

N

- navegación adaptativa, teclado 4-2
- Notas de esta versión 4-27
- números de línea 5-8
 - ir a una línea determinada 5-8
 - mostrar en el editor 5-8
- números de línea para seleccionar texto 5-15

O

- ocultar
 - código 4-2
 - el panel de mensajes 4-11

- opciones 4-22
 - buscar 4-22
 - de importación
 - ClassInsight 6-9
 - definir favoritos 4-22
 - desplazamiento 4-22
 - personalizar 4-22

P

- paleta de componentes 4-1
- panel de contenido 4-23
 - ocultar otros paneles 4-20
 - pestañas de archivo 4-24
 - pestañas de vistas de archivo 4-17
 - redimensionar 4-20
- panel de estructura 4-17
 - Carpeta Por hacer 4-19
 - desplazarse por la estructura de archivo 4-21
 - filtrar 4-17
 - mensajes de error 4-20
 - orden 4-24
 - usar para desplazarse por el código fuente 4-26
- panel de mensajes 4-26
 - copiar texto 2-17
 - integrar 5-20
 - mostrar 4-26
 - mostrar etiquetas todo 4-26
 - ocultar 4-11
 - segregar 4-11
- panel del proyecto
 - barra de herramientas proyecto 4-12
 - buscar 6-4
 - ejecutar aplicaciones 4-12, 4-16
 - filtrar paquetes 4-11
 - menú contextual 4-11
 - pestaña Buscador de archivos 5-6
 - pestaña Proyecto 7-4
- panel fuente, dividir vista 6-10
- paneles, disposición 4-6
 - Consulte* configuración del espacio de trabajo
- paquetes, añadir al proyecto 4-32
- ParameterInsight 6-17
- personalizar
 - color
 - preferencias 7-11
 - elementos de pantalla 4-21
 - elementos de Sync Edit 5-8
 - IDE 6-18
 - opciones del visualizador 5-15
- pestañas 4-24
 - Buscador de archivos, panel del proyecto 5-6
 - del panel del proyecto 7-4

- panel de contenido 4-17, 4-24
- plantillas de código 6-5, 6-20
 - añadir al código 6-23
 - crear 6-5, 6-22, 7-11
 - desplazarse mediante pestañas 6-19
 - editar 6-21
 - importaciones automáticas 6-19
 - insertar 6-5
 - insertar macros de plantilla 5-12
- preferencias
 - color 7-11
- profundizar en la estructura de archivo 4-21, 4-26
- proyectos 7-7
 - abrir existente 4-13
 - añadir aplicaciones 7-11
 - añadir archivos 7-4, 7-10
 - añadir archivos fuente de Java 4-13
 - añadir clases 7-10
 - añadir paquetes 4-32, 7-11
 - cambiar nombre 7-2
 - cerrar 4-13
 - crear 5-4, 7-10
 - de bienvenida 4-27
 - eliminar archivos y carpetas 5-4
 - más información acerca de 7-7
 - ver archivos 7-7
- pulsaciones de teclas 5-19
 - añadir 6-11
 - asignar nuevas 6-11
 - modificar 6-11
 - personalizar 5-19

R

- ratón, navegación alternativa 2-16
- redimensionar panel de contenido 4-20
- reducir el tamaño del texto del código 5-9
- resolver conflictos Javadoc 6-11

S

- ScopeInsight 5-8
 - identificar variables sin utilizar 5-14
- segregar panel de mensajes 4-11
- seleccionar texto mediante números de línea 5-15
- símbolos de código
 - buscar definición 2-17
- símbolos de código.buscar referencias 6-25
- sincronizar tabla de contenido con archivo de ayuda abierto 2-9
- sitio web de Borland Community 4-27
- Sync Edit 5-14
 - editar 5-14
 - personalizar 5-8

T

- TagInsight 5-4
- tamaño de fuente, cambiar en el editor 3-1
- texto
 - buscar 5-13
 - del código fuente (cambiar tamaño) 5-9
 - localizar 5-13
 - seleccionar en el editor 8-1
- tutoriales 4-2
 - creación de aplicaciones 4-2
 - creación de un applet 4-27

U

- Usenet, grupos de noticias 6-4

V

- vistas jerárquicas 4-17
- visualizador
 - abrir varias instancias 4-27
 - ampliar árboles 5-1
 - barras de estado 4-27
 - buscar 4-1
 - contraer árboles 5-1
 - de la ayuda
 - ampliar la vista 2-14
 - añadir pestañas 2-14
 - búsqueda completa de texto 2-9
 - cambiar de panel 2-9
 - desplazarse con el teclado 2-7
 - índice de materias 2-9
 - interfaz de usuario 2-8
 - sincronizar árbol con archivo de ayuda abierto 2-9
 - desplazarse por los paneles
 - árboles de navegación del visor 6-11
 - editor 2-11
 - IDE de JBuilder 2-14
 - ocultar paneles 4-20
 - redimensionar panel de contenido 4-20
- Visualizador de la ayuda
 - buscar texto en tema actual 2-10
 - colocar marcadores 4-3
 - copiar texto 5-11
 - desplazamiento 6-2

