

B. Apéndice B: Introducción a C++ Builder.

B.1. Estructura de C++ Builder.

B.2. C++ Builder.

B.3. Requisitos hardware.

B.4. Instalación de C++ Builder.

B.5. Entorno de desarrollo integrado.

B.6. El gestor de proyectos.

B.7. Opciones del proyecto.

B.8. Archivos fuente generados por C++ Builder.

B. Apéndice B: Introducción a C++ Builder.

En este apéndice se pretende dar una introducción al entorno de desarrollo integrado C++ Builder. Puede ser de utilidad para los no iniciados que quieran comprender en su plenitud el presente proyecto de fin de carrera.

B.1. Estructura de C++ Builder.

C++ tiene la fama de ser un lenguaje muy poderoso, pero con el poder viene la responsabilidad y el programador debe conocer una serie de conceptos y reglas de uso. Esto requiere de suficiente experiencia y conocimiento del lenguaje cuando está programando aplicaciones windows complejas. El hecho es que C++ es un lenguaje complejo con demasiadas reglas y términos confusos, y el usuario tiene además que aprender técnicas de programación orientadas a objetos.

Productos como Borland Delphi y Microsoft Visual Basic, vienen a provocar un cambio irreversible en la programación visual, ya que usando esas herramientas, los programadores pueden crear aplicaciones de forma mucho más fácil y rápida de lo que lo hacían anteriormente, ahora los desarrolladores solo tienen que mover los objetos que requieren usar, modificarlos de acuerdo a sus necesidades (solo llenando las funciones vacías), y eso es todo: se obtiene una aplicación ejecutable.

C++ Builder trabaja en la misma línea que Delphi y Visual Basic. Este desarrollo rápido de aplicaciones hace más fácil la construcción de proyectos. Con estas herramientas de programación usted tiene más poder con menos esfuerzo.

C++ Builder produce aplicaciones ejecutables bajo Windows 95, Windows 98, Windows ME, Windows NT, Windows 2000 y Windows XP con soporte de 32 bits.

Y puede ser que C++ Builder le haga un mejor programador.

B.2. C++ Builder.

Borland ofrece tres diferentes versiones de C++ Builder y cada versión vendrá a satisfacer unas necesidades determinadas:

- **C++ BUILDER STANDAR.** Proporciona todas las herramientas que necesita para el desarrollo de aplicaciones de bases de datos, esta versión ocupa aproximadamente 75 MB de espacio en su disco duro.
- **C++ BUILDER PROFESIONAL.** Ofrece C++ Builder con el equipo de desarrollo y otras herramientas potentes como librerías y código fuente del que puede aprender sirviendo como tutorial. Esta versión proporciona algunas secciones especiales que probablemente no requiera inmediatamente, pero se alegrará al familiarizarse más con C++ Builder, esta versión ocupa cerca de 100 MB de espacio en su disco duro.
- **C++ BUILDER, SERIE CLIENTE/SERVIDOR.** Estamos hablando de compatibilidad con bases de datos muy grandes, esta versión está equipada con desarrollos profesionales que necesitan enlazar con grandes sistemas de Bases de Datos, tomando la arquitectura cliente/servidor. Esta versión ocupa un mínimo de 130 MB de espacio en su disco duro.

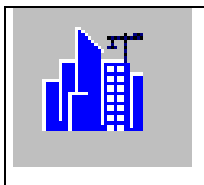
B.3. Requisitos hardware.

Necesita un PC con un procesador lo suficientemente potente y una amplia memoria. Mientras más poderoso sea su PC, más productivo puede ser. C++ Builder es un producto bastante potente y como tal realiza importantes demandas de recursos a su máquina.

Para una correcta ejecución de sus aplicaciones, se recomienda un procesador Pentium y por lo menos 64 MB de RAM. Ya que el desarrollo en C++ Builder corre en sistemas operativos de 32 bits, bajo Windows 95 o Windows NT, la capacidad necesaria de disco duro depende de la versión de C++ Builder que esté usando.

B.4. Instalación de C++ Builder.

Para instalar C++ Builder en su PC, coloque el CD ROM Borland C++ Builder en su lector de CD. No necesita teclear nada, automáticamente el CD comienza la ejecución, emepzando por las preferencias de instalación del programa (Completa, Compacta, Personalizada) y su propia información para registrar el producto, si no se tiene suficiente espacio en disco duro, puedes instalar la versión mínima y tendrá que tener el CD en su lector de disco compacto cada vez que quiera modificar una aplicación.



Al final de la instalación, aparecerá un nuevo grupo de programas, podrás ejecutar la aplicación haciendo doble clic en el icono de C++ Builder.

Fig. B.1: Icono de aplicación de C++ Builder.

B.5. Entorno de desarrollo integrado.

Cuando usted inicia C++ Builder, espera ver una sólo ventana para desarrollar sus aplicaciones; pero C++ Builder le presenta un grupo de ventanas dispersas alrededor de su ventana principal (Fig. B.2). Al hacer doble clic en el icono de C++ Builder se obtiene algo así:

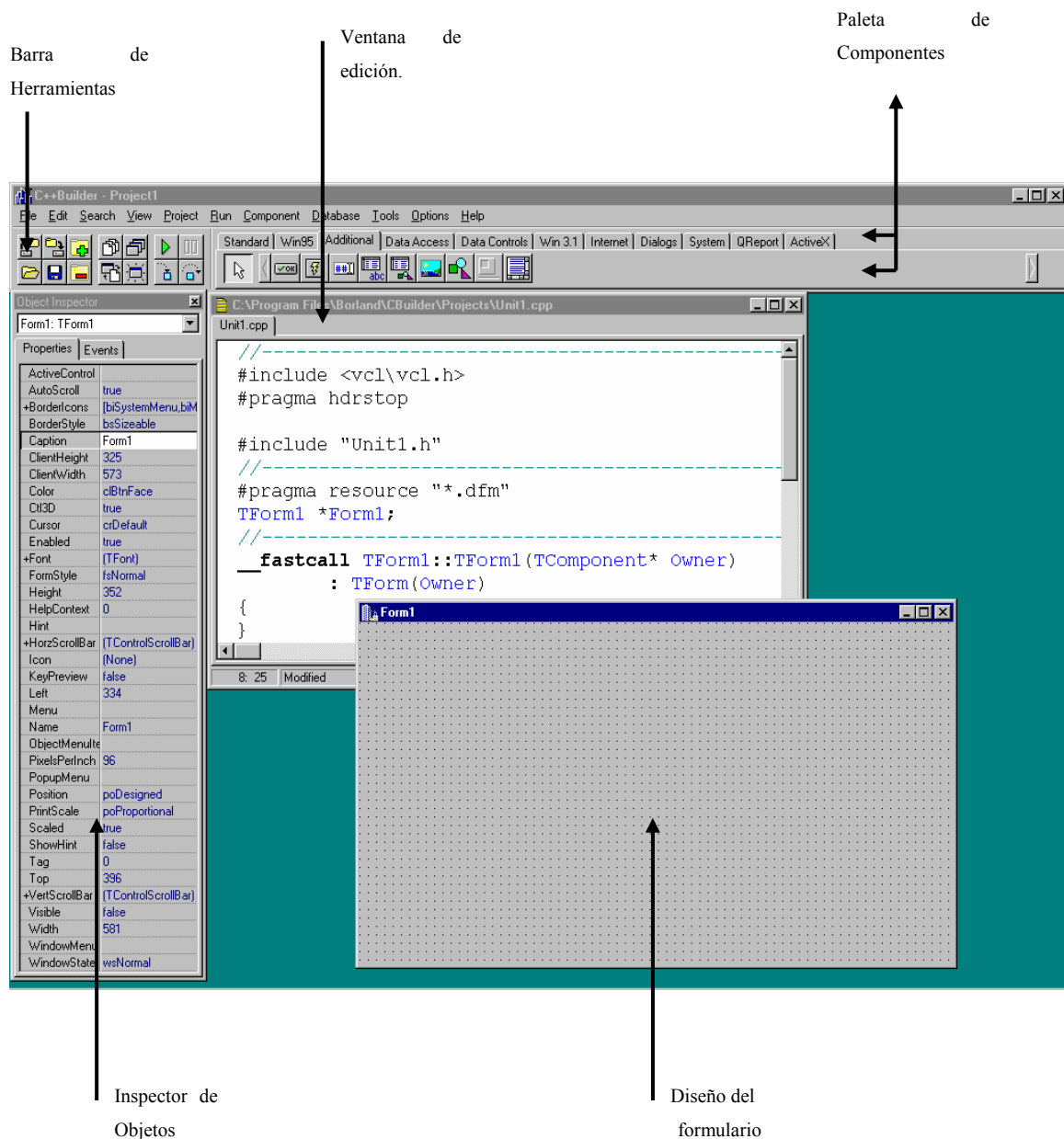


Fig. B.2: Entorno de desarrollo integrado C++ Builder.

En la figura se presentan los principales elementos de C++ Builder (Entorno de Desarrollo Integrado, por sus siglas en inglés IDE). Cada parte en el entorno de desarrollo trabaja conjuntamente: diseños visuales y editor de código. En éste la edición

es similar a otros editores; solo que con el entorno de desarrollo integrado, usted puede observar realmente lo que esta construyendo en el momento de crearlo.

Idealmente se desearía trabajar con una resolución de 800 x 600 o tal vez mayor en su monitor, ya que estas resoluciones dan una sensación de amplitud; pero recuerde que al diseñar sus aplicaciones el usuario final puede trabajar con una resolución diferente a la suya.

B.5.1. Menú principal y barra de herramientas.

Muchas de las opciones a las que puede acceder desde el menú principal, están disponibles a través del panel de botones rápidos en la barra de herramientas. En general la barra de herramientas proporciona una manera rápida de ejecutar operaciones del programa con un simple clic de su ratón.

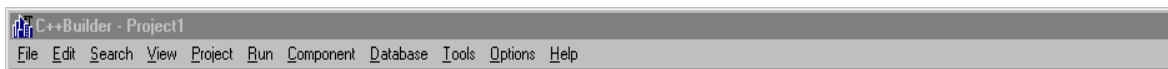


Fig. B.3: Menú principal.



Fig. B.4: Barra de herramientas.

Usted puede configurar la barra de herramientas, seleccionándola con un clic, y nuevamente haciendo clic con el botón derecho del ratón, se activará un menú, y al elegir la opción de propiedades, tendrá a su disposición el editor de la barra de herramientas donde podrá escoger las propiedades que necesite para añadirla a la barra de herramientas solamente cogiéndola y arrastrándola con el ratón al panel de la barra de herramientas.

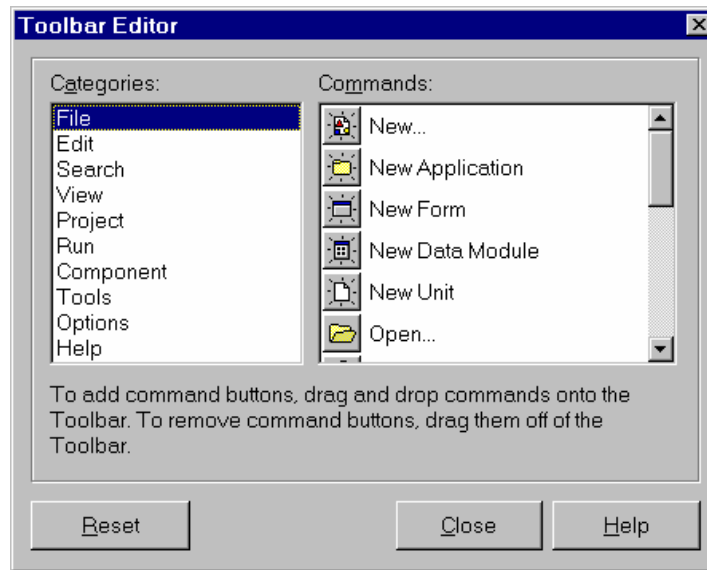


Fig. B.5: Editor de la barra de herramientas.

Si lo que desea es quitar alguna propiedad de la barra de herramientas, solamente arrástrela con el ratón fuera del panel de la barra de herramientas y desaparecerá.

B.5.2. Paleta de componentes.

La paleta de componentes es algo así como un catálogo de objetos que puede usar de acuerdo a las necesidades de construcción de sus aplicaciones. Está dividida en páginas o grupos de acuerdo a sus funciones. Para implantar uno de estos componentes en su aplicación, solo tiene que seleccionarlo con el ratón haciendo un clic en el objeto deseado y hacer clic en el formulario principal (formulario de edición, ventana punteada) para que ya pueda utilizar ese objeto. C++ Builder soporta docenas de componentes.



Fig. B.6: Paleta de componentes.

Standard: Esta tabla contiene los objetos para hacer eficaces y elegantes sus aplicaciones Windows, incluye componentes para desplegar y editar texto, botones, barras de estado y menús.



Fig. B.7: Paleta de componentes Standard.

Win95: Estos componentes permiten el acceso a los controles de usuario-interfaz de Windows 95. Uno de los principales es la vista del árbol de directorio (conocido como windows explorer), control de página, etc.



Fig. B.8: Paleta de componentes Win95.

Additional: La tabla de adicionales contiene algunos de los mejores y variados de la paleta de componentes, como mapas de bits, botones rápidos y componentes de apariencia.



Fig. B.9: Paleta de componentes Additional.

Data Access y Data Controls: Se puede acceder a bases de datos y hacer consultas dentro de las aplicaciones que construya con las facilidades que permiten estos dos grupos de objetos.

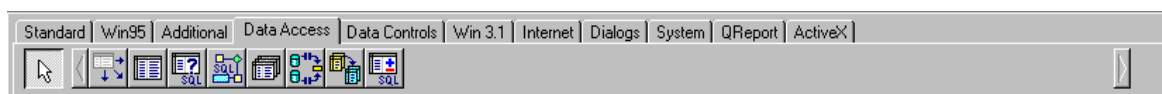


Fig. B.10. Paleta de componentes Data Access.



Fig. B.11: Paleta de componentes Data Controls.

Win31: Muchos de los controles en Win31 tienen equivalentes en Win95 pero estos pueden usarse para dar sentido a aplicaciones en windows 3.1, además de proporcionar un bloc de notas.

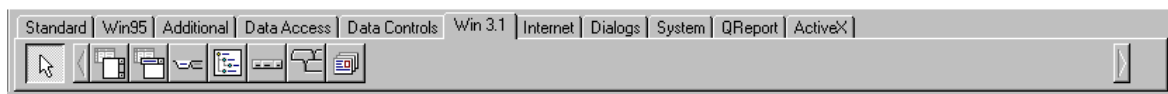


Fig. B.12: Paleta de componentes Win 3.1.

Internet: Esta tabla dada por C++ Builder, comprende lo referente al grupo de herramientas de internet.

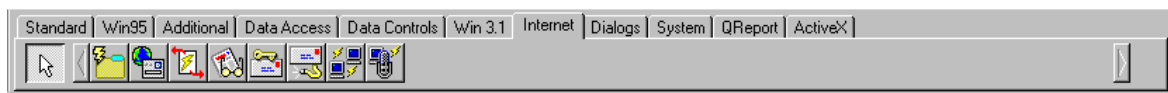


Fig. B.13: Paleta de componentes Internet.

Dialogs: Permite hacer cuadros de diálogo que agilizan el desarrollo de sus aplicaciones como el abrir y grabar archivos, seleccionar tipos de letras, colores e impresoras y mucho más.

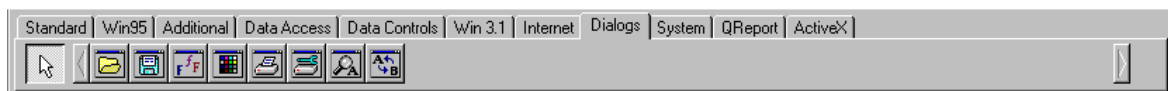


Fig. B.14: Paleta de componentes Dialogs.

System: Proporciona controles individuales para seleccionar archivos, directorios o unidades.



Fig. B.15: Paleta de componentes System.

Qreport: (o Quick Reports) proporciona componentes para que pueda fácilmente organizar sus informes y presentar una vista preliminar.

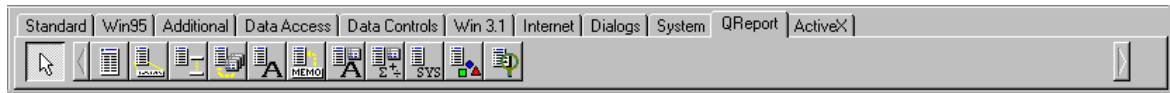


Fig. B.16: Paleta de componentes Qreport.

ActiveX: Esta tabla de componentes contiene un comprobador de ortografía así como objetos gráficos impresionantes.

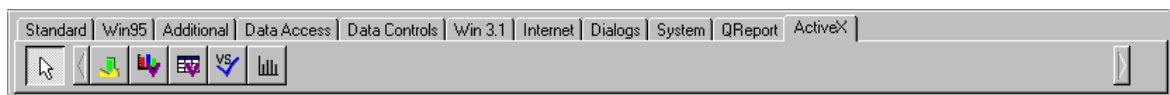


Fig. B.17: Paleta de componentes ActiveX.

Durante el desarrollo de este trabajo, solo presentaré el uso de algunos de los objetos de la tabla de componentes, pero la idea es la misma para todos los objetos de esta paleta.

B.5.3. El editor de formulario.

Cuando comience a trabajar con C++ Builder, su espacio de trabajo y resultados estará constituido por el formulario principal.

Cada formulario representa una ventana individual en su aplicación, en el formulario puede diseñar, añadir, eliminar y reconfigurar los componentes según las necesidades de su aplicación.

B.5.4. El inspector de objetos.

El inspector de objetos permite ver las propiedades o características de los objetos que conformen su proyecto, por medio de él se pueden cambiar las propiedades de los objetos, también muestra los eventos asociados a dichos objetos.

Cuando se selecciona un objeto, el inspector de objetos automáticamente cambia al contenido y propiedades de este objeto. Si se oculta, se puede volver a visualizar pulsando la tecla de función F11.

El inspector de objetos podría llamarse “editor de objetos”, por la propiedad antes mencionada de poder modificar las propiedades de los objetos.

Propiedades: Cuando se comienza un proyecto el inspector de objetos despliega las propiedades del formulario principal como son: nombre, color, altura, ancho, posición etc.

Recordemos que al seleccionar otro objeto, automáticamente mostrará las propiedades de ese objeto.

Eventos: La tabla de eventos despliega para cada objeto los eventos como son: Al activar el objeto, al presionar una tecla, al presionar el ratón, al soltar el ratón, etc. esos eventos son lanzados con acciones del usuario, o del propio sistema operativo mismo. Por ejemplo, el evento “*al hacer clic en el ratón*” se dispara o actúa cuando el usuario hace clic con el ratón en ese objeto.

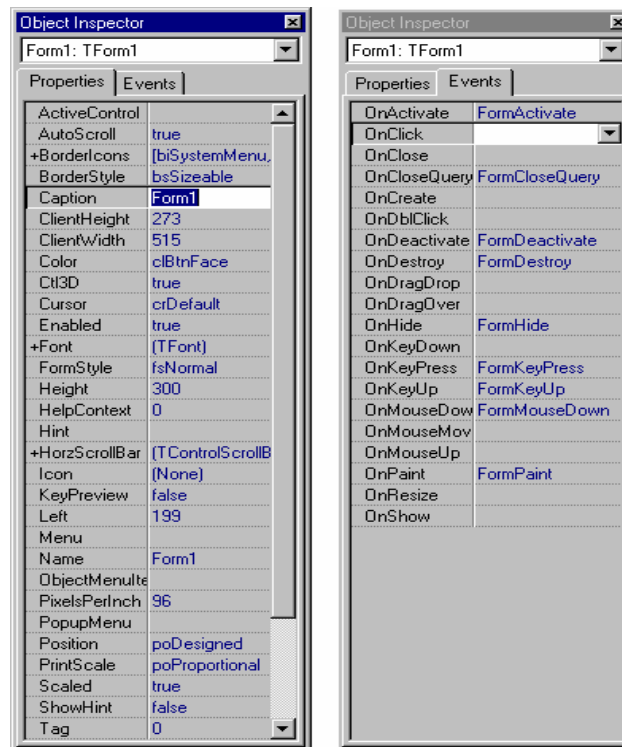


Fig. B.18: Inspector de objetos.

B.5.5 El editor de código.

La ventana de edición de código muestra el código actual de tu aplicación C++ Builder.

Al añadir objetos y hacer doble clic sobre ellos, automáticamente se editará en la ventana de edición la llamada a la función que asociará al evento de ese objeto, dejando el espacio en blanco para que se escriba el código correspondiente a la acción que se desee para ese evento.

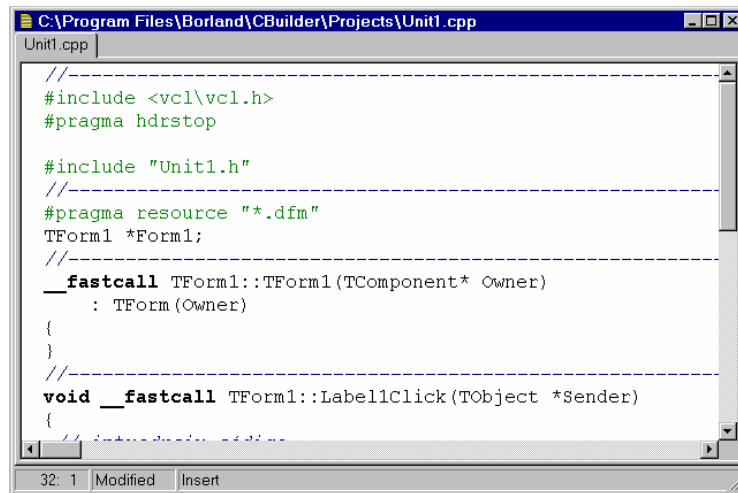


Fig. B.19: Ventana de edición de código.

B.6. El gestor de proyectos.

Un sencillo proyecto de C++ Builder está formado por sólo un formulario y su código, pero en aplicaciones muy grandes, puede formarse un proyecto con varios formularios, código y varios archivos de cabecera distintos a las librerías que por omisión ya necesita la aplicación, por tanto un proyecto puede integrar varios archivos. Para saber cuales son los archivos que componen un proyecto, usaremos el gestor de proyectos, que muestra el árbol de archivos involucrados en el orden en que fueron añadidos.

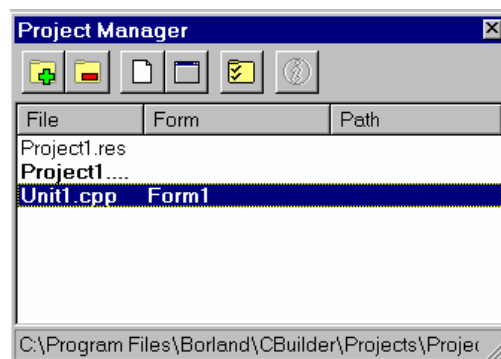


Fig. B.20: Gestor de proyectos.

B.6.1. Como añadir archivos a un proyecto.

Para añadir archivos a un proyecto, seleccione del menú principal: View ⇒ Project Manager ⇒ y al hacer doble clic obtendrá una caja con la información de los componentes actuales del proyecto (**Fig. B.21**), haga clic en el botón de integración al

proyecto, , y saldrá una ventana de adición “Add to project”, donde puede buscar la ruta de acceso donde se encuentre el archivo a añadir, y haga clic en el botón “Ok” para cerrar el cuadro de diálogo y obtendrá ya añadido a su proyecto el archivo seleccionado.

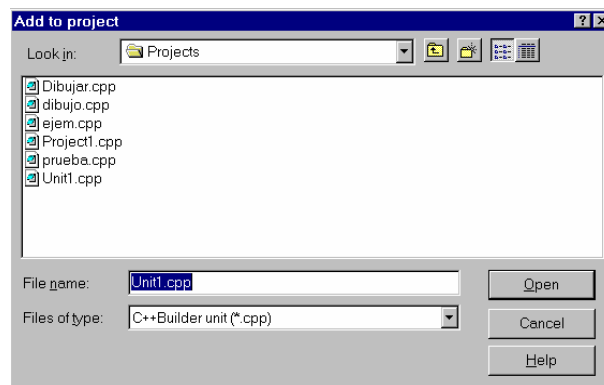


Fig. B.21: Ventana de adición al proyecto.

B.6.2. Como eliminar archivos de un proyecto.

Para eliminar un archivo del proyecto, abra su gestor de proyectos, seleccione el archivo que quiera eliminar y haga clic en el botón de borrar archivo  y será borrado de la lista (no de su máquina, sólo del proyecto).

B.7. Opciones del proyecto.

Puede acceder a detalles del entorno de configuración del proyecto actual eligiendo del menú principal Options $\Rightarrow$ Project.

Este comando abre el cuadro de diálogo de opciones del proyecto, toma el entorno para cada proyecto y puede acceder a él en cualquier momento durante el desarrollo de la aplicación.

Puede explorar las seis hojas para ver las opciones que están disponibles. Al iniciar un proyecto, comienza con la configuración que tiene por defecto.

B.7.1. Página de formularios.

Por omisión está incluida la auto creación de un formulario, al abrir un nuevo proyecto, evitando que usted al crear una nueva aplicación tenga que pedir una formulario. Al crear un nuevo proyecto se crea automáticamente su formulario principal y su ventana de edición de código.

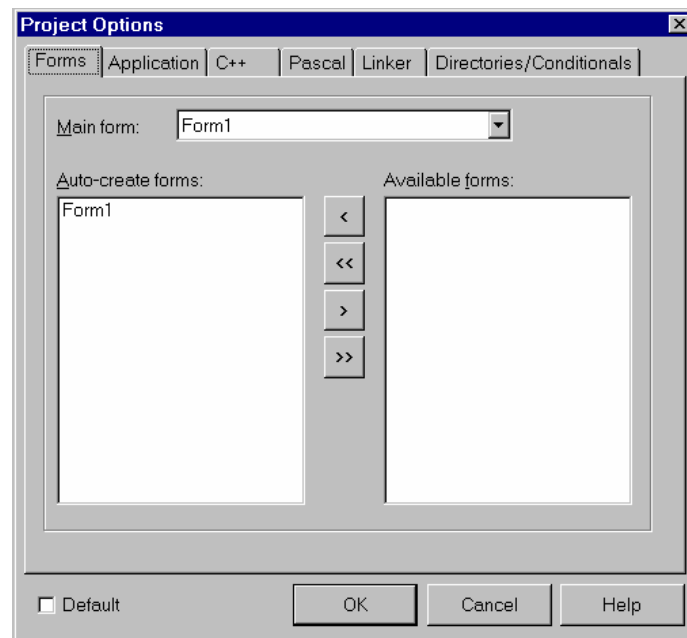


Fig. B.22: Opciones del proyecto, página de Formularios.

B.7.2 La página de aplicación.

La página de aplicación contiene tres entradas de configuración:

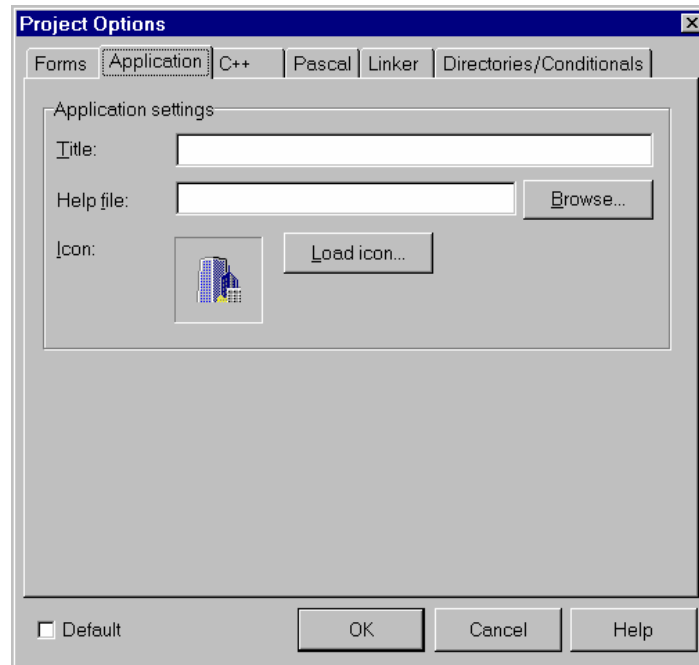


Fig. B.23: Opciones del proyecto, página de Aplicaciones.

Título: El texto que introduzca será el título de la aplicación y será desplegado con el icono cuando se minimice la aplicación.

Help_File: Asociará un archivo de ayuda a su aplicación.

Icon: El archivo ejecutable contendrá el icono que seleccione, por omisión mostrará el de aplicaciones C++ Builder.

B.7.3. La página de C++.

La página C++ es donde inicializa las opciones que desea usar para configurar su proyecto particular, como por ejemplo, opciones de depuración, optimización de código y librerías precompiladas.

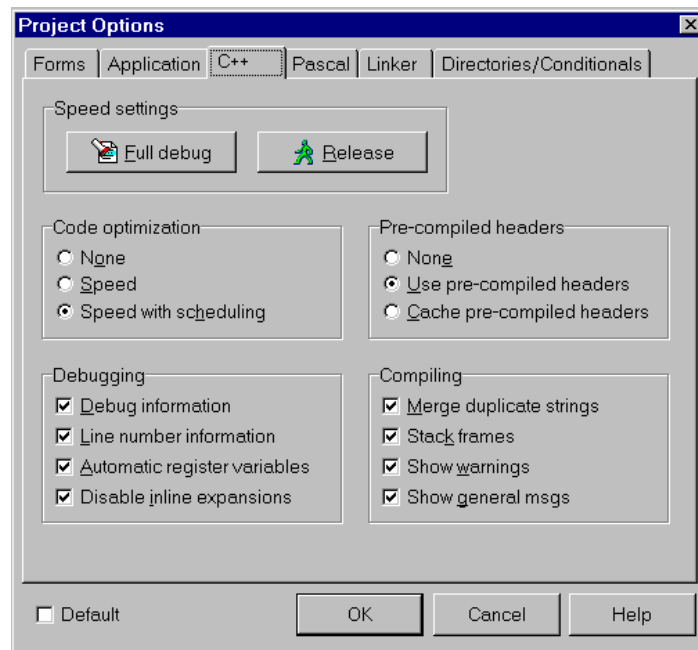


Fig. B.24: Opciones del proyecto, página de C++.

B.7.4. La página Pascal.

C++ Builder tiene un hermano mayor llamado Delphi que es una herramienta RAD. Y sus aplicaciones se construyen en lenguaje Pascal. Si ha manejado Delphi, verá que el entorno de desarrollo es prácticamente el mismo.

Puede usar funciones hechas en Delphi dentro de C++ Builder.

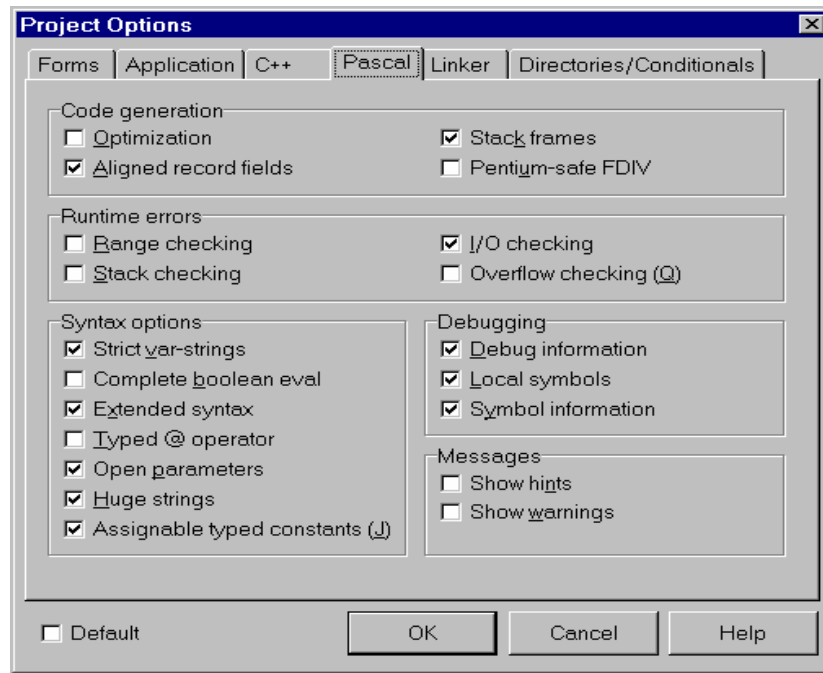


Fig. B.25: Opciones del proyecto, página de Pascal.

B.7.5. La página del linkador.

En la mayoría de las ocasiones querrá que la aplicación genere un archivo ejecutable, en otras ocasiones querrá que su construcción sea una librería para diferentes programas, en este caso usted puede llamar a la opción de librerías de enlazado dinámico (DLL) y las puede generar cambiando la opción a generar DLL's.

B.7.6. Página directorios/condicional.

Usted no necesita cambiar la configuración en esta página, a menos que use diferentes directorios de los que por omisión genera la instalación de C++ Builder. Y debe tener cuidado en estos parámetros.

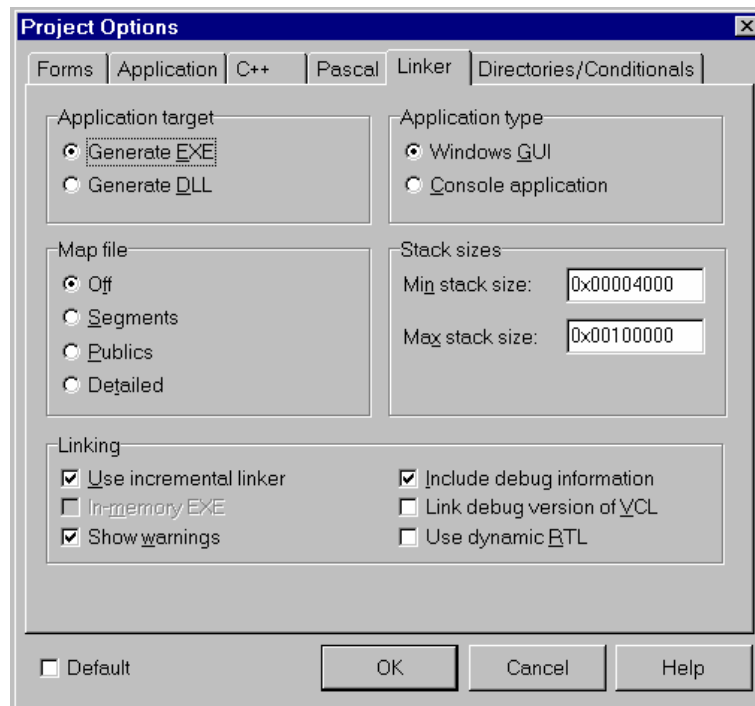


Fig. B.26: Opciones del proyecto, página Linker.

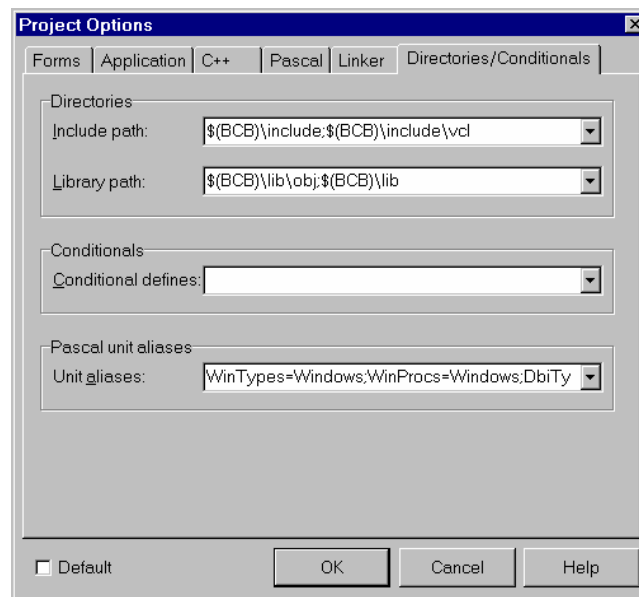


Fig. B.27: Opciones del proyecto, página de directorios/Condicional.

B.8. Archivos fuente generados por C++ Builder.

Cada una de las extensiones que vamos a ver son vitales para cualquier proyecto de C++ Builder.

Los siguientes archivos contienen detalles del diseño de sus proyectos y formularios, tenga cuidado en no perder ninguno ya que puede necesitarlos si quisiera modificar alguna aplicación hecha.

Project1.mak: Este es el archivo principal de opciones del proyecto. Un archivo .mak se requiere en cada aplicación; es un archivo de texto que puede examinar eligiendo del menú principal: View $\Rightarrow$ Project MakeFile. Este archivo contiene instrucciones de cómo C++ Builder construirá el archivo (.exe) ejecutable para el proyecto.

Project1.Cpp: Este archivo contiene el código principal de la aplicación, comparte el mismo nombre del proyecto, lo crea automáticamente C++ Builder al darle nombre al proyecto; contiene el código de iniciación y terminación del programa. Si quiere examinar este archivo, sólo elija del menú principal View $\Rightarrow$ Project Source.

Unit1.cpp: Este es el código que usted le da a los eventos de los objetos que tendrá en su formulario final, este código es el que introduce en la ventana de edición, o en su editor de código.

Unit1.h: Para cada archivo .cpp, C++ Builder crea automáticamente un .h correspondiente. El archivo de cabecera contiene la declaración del formulario e indica a C++ Builder la lista de componentes y los eventos que tendrá la aplicación.

Unit1.dfm: El archivo .dfm contiene la información, definición y declaración del formulario y otros detalles importantes como: tamaño, color, títulos, fondos etc. así como detalles del resto de los componentes utilizados en el formulario.

La extensión .dfm indica que este archivo oculta los datos del formulario en formato binario. Este archivo no se puede leer, pero puede convertirlo para observar su contenido, solo seleccione el formulario, con el botón derecho del ratón haga clic y saldrá un menú de opciones, escoja “View as text”, para ver la información de su formulario.

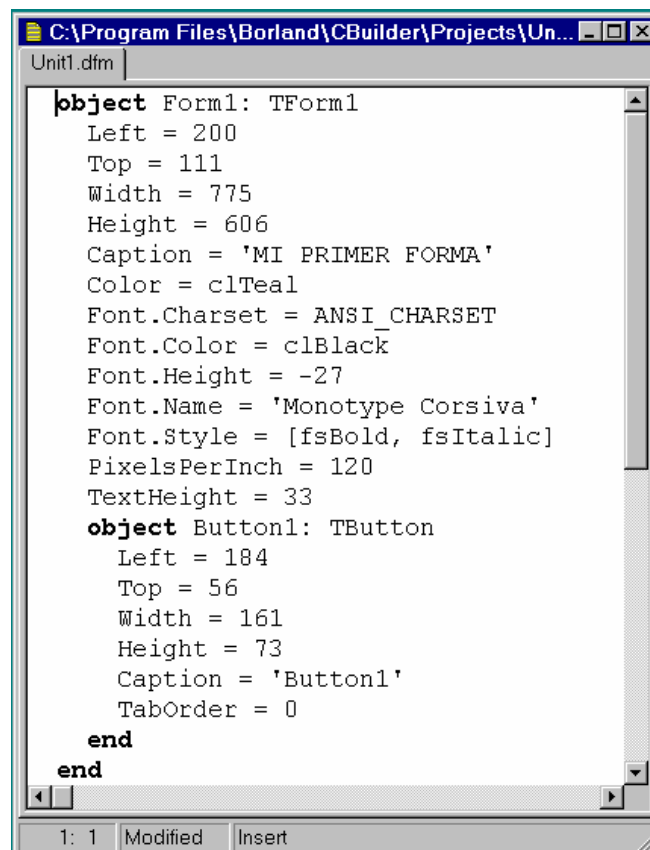


Fig. B.29: Ver el formulario como texto.

De igual manera, para volver a verlo como formulario, seleccione con el botón derecho del ratón, escoja “View as Form”, y volverá a su formulario.

Unit1.obj Cuando se compila el proyecto se crea el archivo binario con extensión .obj. Cada aplicación contendrá un archivo .obj, este archivo cambiará cada vez que se reconstruya el proyecto.

Project1.exe o Project1.dll: Este es el archivo final según se haya elegido en su proyecto, el .exe podrá ejecutarse, y los .dll son librerías dinámicas que pueden utilizarse desde otros programas.

Project1.dsk: Contiene la información de la configuración que tenía su hardware en el momento de crear o finalizar su proyecto, para que cuando vaya a ejecutarse lo haga sobre la misma configuración.

Project1.il?: La extensión .il? indica que el archivo es usado al enlazar, C++ Builder usa una tecnología de compilación muy rápida, lo que hace que la compilación o recompilación de sus proyectos sea realice en cuestión de segundos.