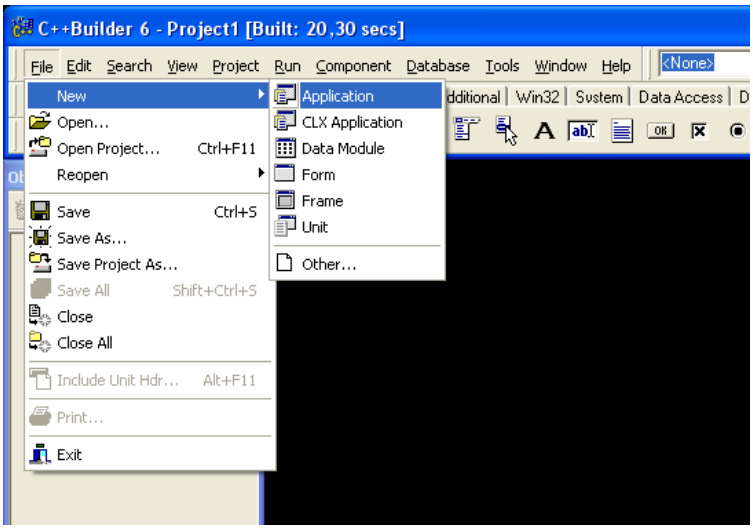
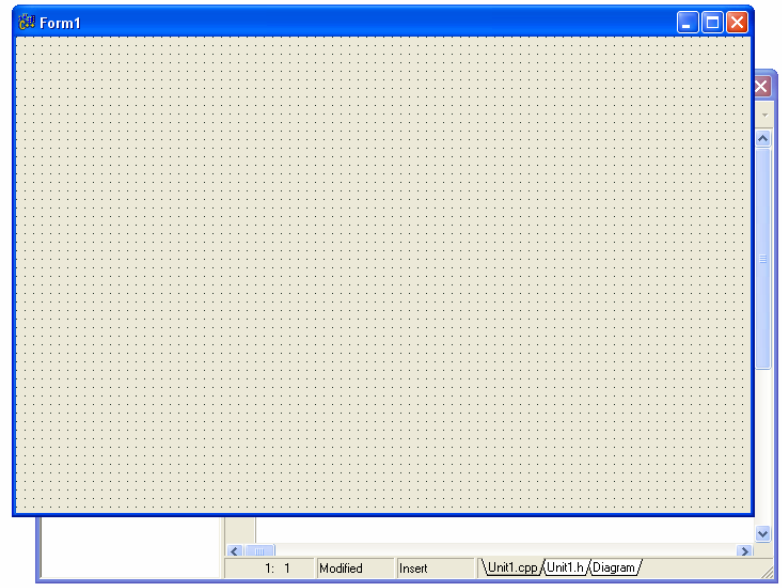
Primeros pasos con Builder

En este repaso introducimos los conceptos básicos en el uso del Builder. Para empezar crearás una aplicación desde el menú File->New Application (véase figura):



Una vez hecho, se genera un proyecto nuevo al que Builder le da un nombre por defecto (véase el nombre 'Project1' en la barra del título de la figura anterior).

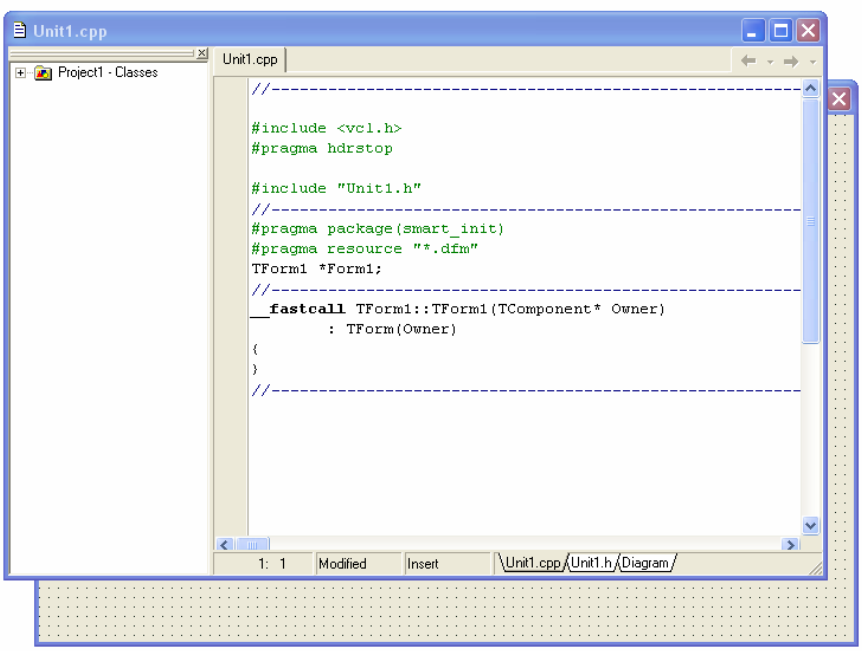
El nuevo proyecto carga un formulario vacío que tiene el aspecto de una ventana de Windows (véase siguiente figura):



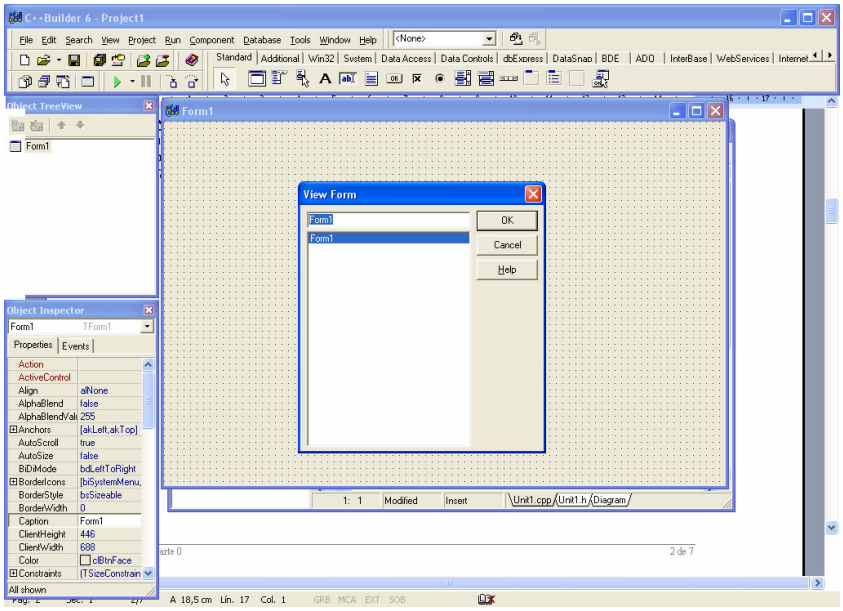
Dicho formulario es un objeto del proyecto. Un proyecto en Builder está compuesto por objetos gráficos (formularios) que son el interfaz de nuestra aplicación y de unidades (bloques en los que se escribirá el código). Builder por defecto asigna un nombre al formulario, en la figura anterior puede verse que el nombre asignado es: 'Form1'.

Recuerda que un proyecto es = al formulario (parte visible) + las unidades (el código de la aplicación). Para cambiar el foco entre estos dos objetos utiliza F12 (pruébalo!).

Cuando el formulario tiene el foco (está en primer plano) y pulsamos F12, el formulario pasa al fondo y se muestra en primer plano la unidad del proyecto. En la siguiente figura puedes ver que el objeto 'Unit' tiene el foco y el formulario 'Form1' queda en segundo plano justo detrás:



Nota: Si en el proyecto hay más de un formulario y accidentalmente has 'perdido' alguno, basta con utilizar Shift+F12 para mostrar la lista de formularios asociados al proyecto actual. Tan sólo hay que doble clic en View/Forms… y hacer doble clic en el formulario buscado:



Eventos

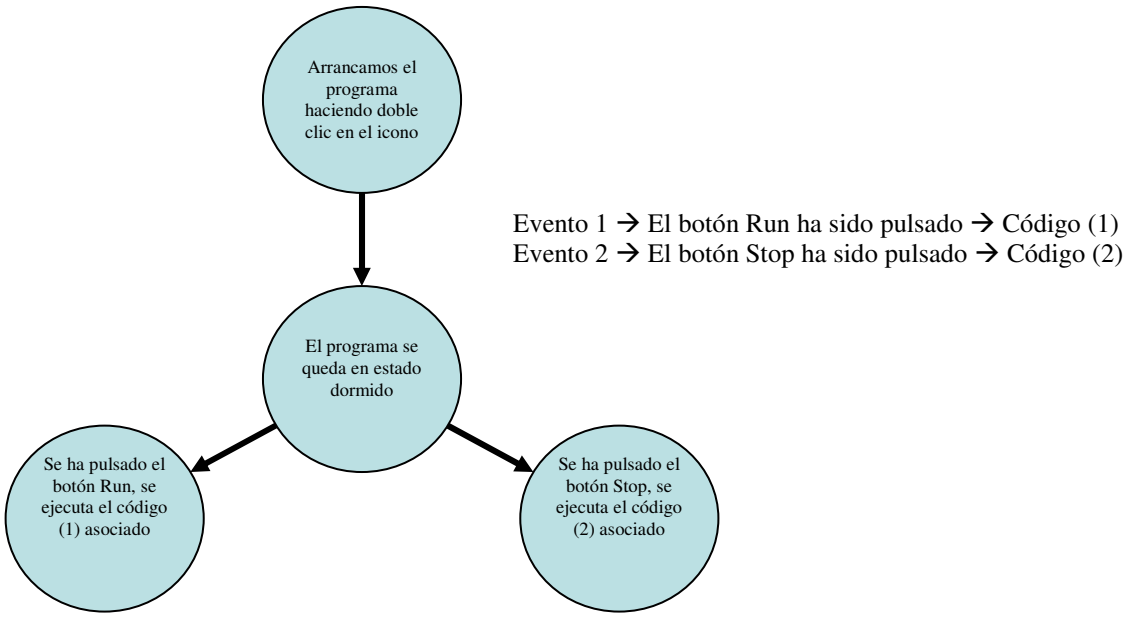
Nuestra aplicación estará controlada por eventos. Esto quiere decir que su única función será esperar a que suceda algo, por ejemplo una pulsación de una tecla, un clic del ratón sobre un botón, etc y mientras no se dé este caso, el programa se quedará en un estado ocioso o 'durmiendo'.

Con esta idea en mente, en la unidad del proyecto tendremos que escribir el código asociado a cada evento (pulsación de tecla, clic del ratón, selección de una lista desplegable, etc…), es decir, todas aquellas que nuestro programa atienda.

Ejemplo: Imagina una aplicación que hace las funciones de temporizador: Dos botones, uno de puesta en marcha 'Run' y otro de parada 'Stop'. En este caso nuestro programa atenderá a dos eventos:

1. Evento 1 →El botón Run ha sido pulsado.
2. Evento 2 →El botón Stop ha sido pulsado.

Una vez definidos los eventos, hay que escribir el código correspondiente a cada evento según la funcionalidad del programa. Resumiendo, el esquema del programa será el siguiente:



Fíjate que el código asociado a cada evento será diferente en función del evento. En el ejemplo del temporizador, el código asociado al botón 'Run' lo pondrá en marcha y el código asociado al botón 'Stop' lo detendrá.

De manera general, una aplicación Builder funcionará de este modo: Definiremos qué eventos supervisará y escribiremos el código asociado a cada uno de ellos y mientras ninguno de los eventos se activen, el programa quedará 'dormido' esperando.

Modos de supervisión

En el caso del cronómetro, la aplicación se comporta de forma pasiva y el usuario es el único que puede 'despertar' a la aplicación. El usuario marca el ritmo del cambio de estados. Pero ¿Qué ocurriría si quisiera arrancar y detener automáticamente el cronómetro con sensores de presencia conectados al puerto usb?

Según lo visto, añadiríamos el código necesario para leer el valor del sensor por el puerto usb y cada vez que el usuario hiciera clic en el botón, el evento ejecutaría el código que leería el sensor y en función del estado activaría el cronómetro.

Ocurre que esta solución tiene un problema, imagina que hay que controlar continuamente el sensor, cada segundo, por ejemplo. El caso que se plantea es:

- El usuario deberá hacer clic cada segundo en un botón para disparar el evento que lee el sensor y controla el cronómetro…

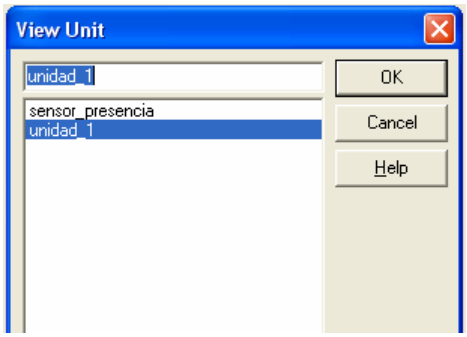
Por supuesto, la opción de tener al usuario enganchado al ordenador haciendo clic en el mismo botón cada segundo, parece, por decirlo del mejor modo, poco eficiente, ¿no te parece?

Para lograr que la aplicación realice una tarea periódica, necesitamos que se dispare un evento y Builder dispone del Timer. Este objeto, parecido a una alarma, permite despertar a la aplicación cada cierto tiempo y realizar una tarea concreta. El tiempo será configurable en ms.

En estos casos podemos decir que la aplicación tiene un modo de supervisión pasiva porque para realizar una tarea concreta no necesita de la intervención del usuario.

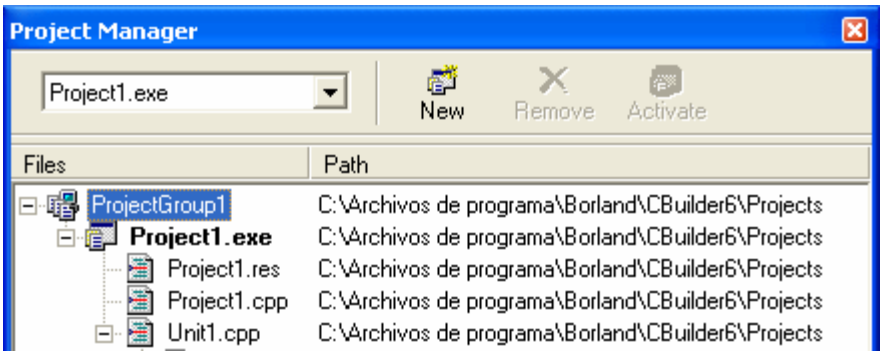
Unidades del proyecto

Las unidades del proyecto son los bloques donde se escribe el código que se va a ejecutar cuando se active los eventos (ratón, teclado, temporizador…). Para mostrar en una ventana todas las unidades del proyecto utilizaremos CTRL+F12.

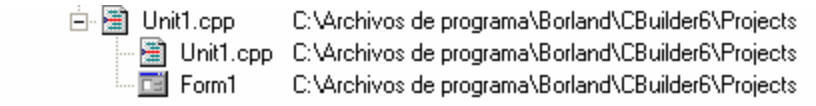


Mostrar todas las unidades del proyecto → CTRL + F12
Mostrar todos los formularios del proyecto → Mayúsculas + F12

Podemos ver la estructura general de un proyecto utilizando el menú View/Project Manager (Ctrl+Alt+F11):

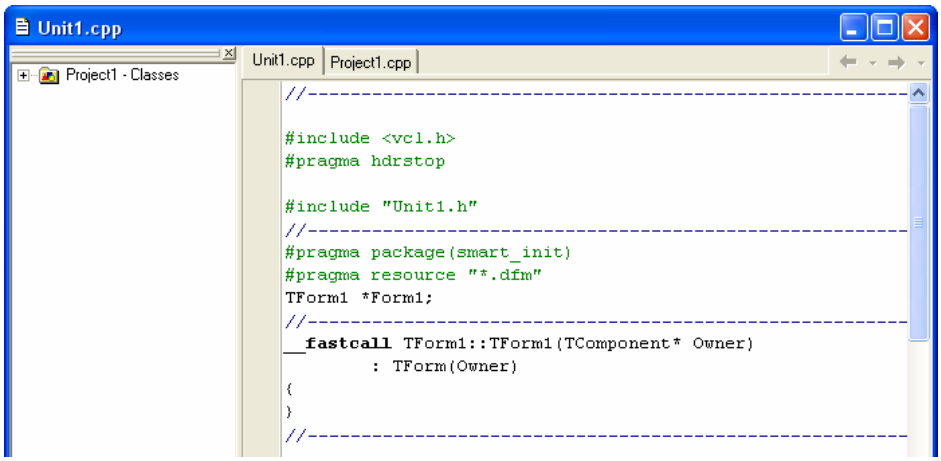


En la ventana anterior podemos ver que el proyecto 'Project1' está formado por el programa principal 'Project1.cpp' y la unidad 'Unit1' y si hacemos clic en el signo '+' que aparece a la izquierda de la unidad vemos que objetos forman la unidad:



Puedes ver en la figura anterior que la unidad 'Unit1.cpp' está relacionada con el formulario 'Form1'. Esto significa que para cualquier evento asociado con el formulario, el código relacionado se escribirá en la unidad "Unit1".

Ahora echemos un vistazo al código que contiene la unidad 'Unit1':



Puedes ver la directiva **#include <vcl.h>** al principio que indica que esta unidad puede utilizar funciones externas de librería necesarias para generar aplicaciones bajo Windows. En proyectos grandes, las funciones se separan por categorías en unidades diferentes para garantizar una mejor gestión y organización. También puedes ver definida una variable Form1 del tipo *TForm1 necesaria para utilizar el formulario.

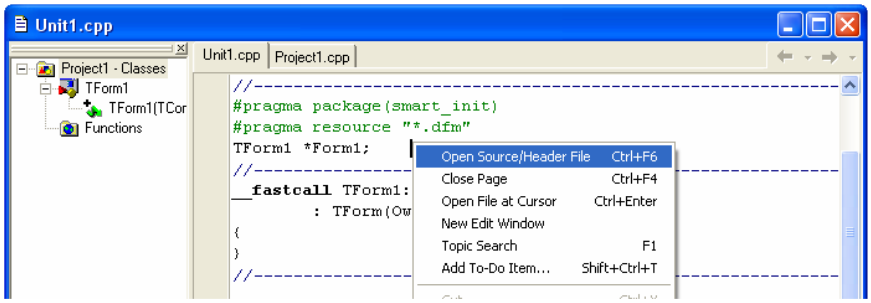
Repasando lo visto hasta ahora, una aplicación se compone, al menos, de un archivo de código con el programa principal y opcionalmente de un conjunto de módulos adicionales en los que se escriben diversas funciones. Builder utiliza el programa principal para almacenar el código del proyecto que normalmente no será visible. Además del código del proyecto, cualquier aplicación contará con al menos un formulario que tendrá vinculado un módulo de código. Para cada formulario adicional, existirá un módulo de código vinculado. También es posible tener otros módulos (unidades) no vinculados a ningún formulario que contendrás otras funciones u objetos necesarios para la aplicación.

Archivos de cabecera_____

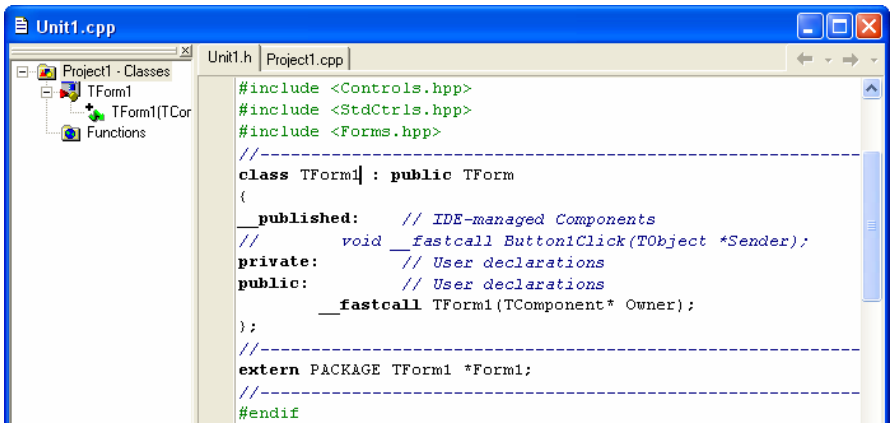
Cuando escribimos funciones en una unidad, los nombres de las mismas quedan registrados en un archivo de cabecera llamado nombre_fichero.h. En términos generales en los archivos de cabecera quedan definidas todas las funciones que deseamos hacer públicas y a las que otras unidades tendrán acceso.

Por ejemplo para mostrar el archivo de cabecera de la unidad Unit1.cpp, sigue lo pasos:

1. Sitúa el cursor en la zona del código de la unidad.
2. Activa el menú contextual → Open Source/Header File (Ctrl+F6)



Podrás ver que en las pestañas de la parte superior del código ahora aparece 'Unit1.h' en vez de 'Unit1.cpp'. Ahora estás viendo el contenido del archivo de cabecera:

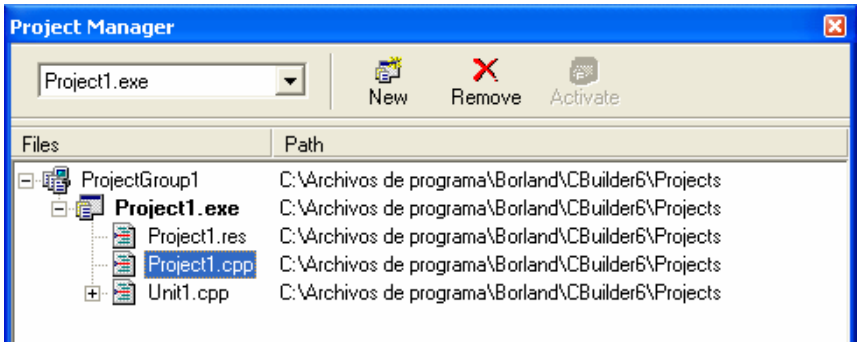


Fíjate que también puedes cambiar entre el archivo de cabecera 'Unit1.h' y la unidad 'Unit1.cpp' utilizando las pestañas mostradas en la parte inferior de la zona de código.

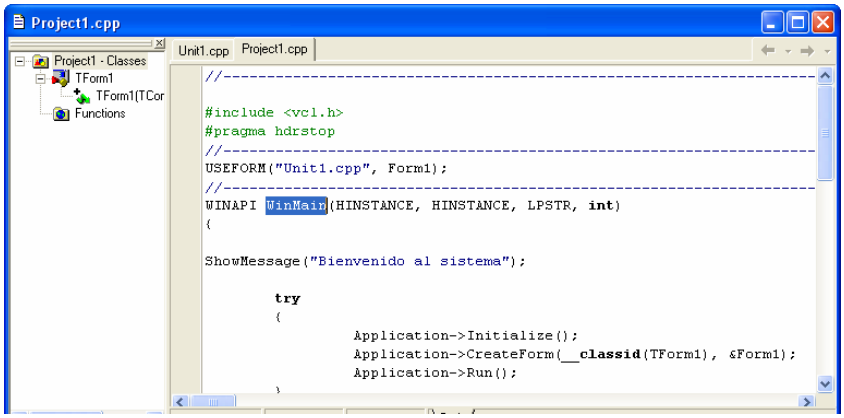
Todo empieza en el programa principal

Cuando se ejecuta una aplicación, su punto de entrada es la función WinMain que está en el programa principal, es decir, en 'Project1.cpp'. Vamos a echarle un vistazo:

1. Selecciona el menú View →Project Manager, verás la siguiente figura:



2. Haz doble clic sobre 'Project1' para ver el código del programa principal WinMain:



Como ejercicio añade el código necesario para mostrar un mensaje por pantalla en la función WinMain como puede verse en la figura:

→ ShowMessage("Bienvenido al sistema");

Guardar el proyecto

Es importante que grabes el proyecto nada más empezar:

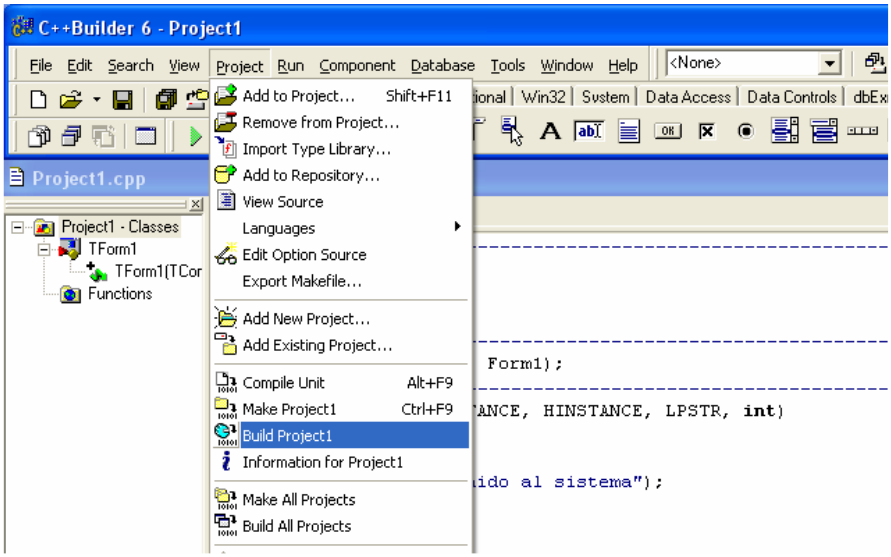
Guardar el proyecto y los archivos
File->Save All (Shift+Ctrl+S)

Crear un directorio para el proyecto

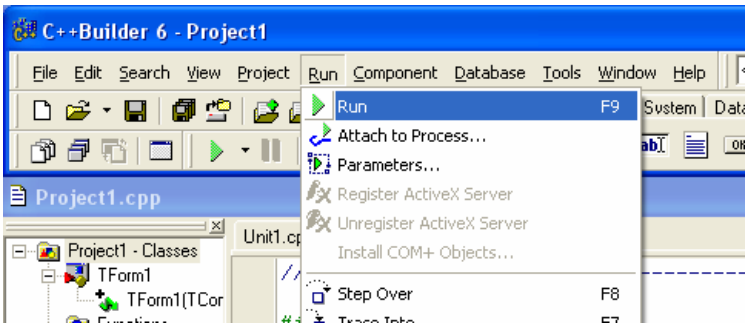
Dar nombre a la unidad Unit1
Dar nombre al proyecto Project1

Compilar y ejecutar una aplicación

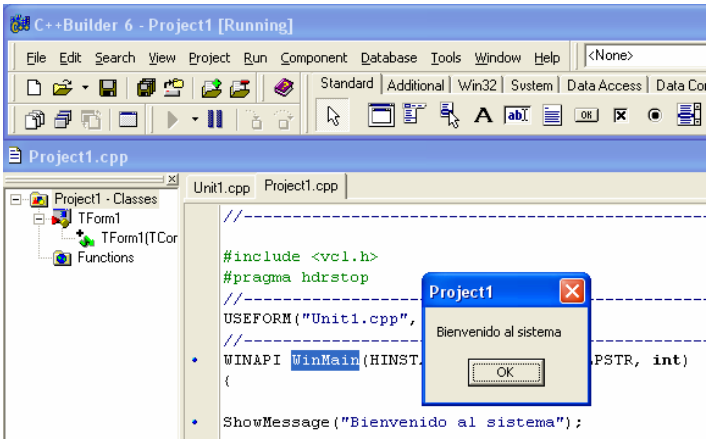
Compilar la aplicación: Project →Build Project



Ejecutar la aplicación: Run →Run (F9)



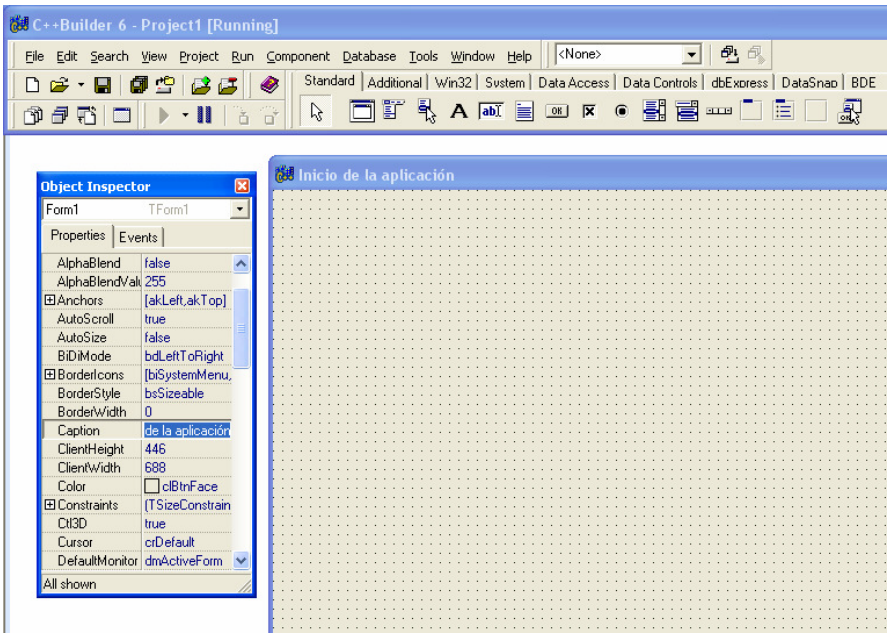
Cuando ejecutes la aplicación, verás que se muestra el mensaje que hemos añadido:



Una vez hagas clic en el botón se cargará el formulario vacío.

Ahora vamos a cambiar algunas propiedades del formulario. Para ver sus propiedades, primero haz clic en cualquier zona del formulario y fíjate en la columna de la izquierda 'Object Inspector'. En esa columna aparecen todas las propiedades que pueden cambiarse del objeto seleccionado, en este caso, un formulario.

Nota: Si no ves el 'Object Inspector' pulsa F11 o el menú Viev/Object Inspector.



Puedes ver dos pestañas, 'Properties' y 'Events'. Si quieres cambiar el nombre del formulario, su tamaño, el color, tipo de bordes, posición, etc…debes hacerlo desde la pestaña propiedades. Cuando tengas que definir un comportamiento sobre un objeto como un clic de ratón, una tecla pulsada, etc debes hacerlo desde eventos.

Modificar las propiedades del formulario_____

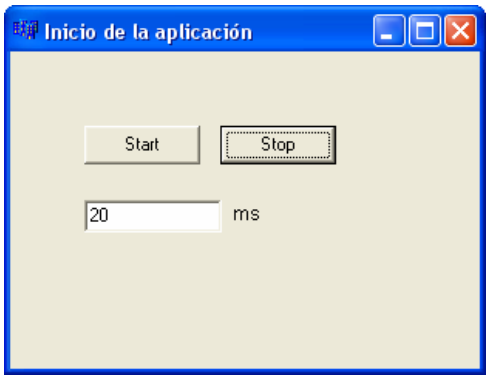
Modificar la propiedad Caption
Modificar la propiedad Name
Modificar la propiedad Color
Modificar la propiedad Width
Modificar la propiedad Height

Ejecutar y ver los cambios
Editar de nuevo

Modificar la propiedad Left
Modificar la propiedad Top
Modificar la propiedad FormStyle a fsStayOnTop (siempre visible)

Ejecutar y ver los cambios
Guardar los cambios
File->Save All (Shift+Ctrl+S)

Se propone la creación de un sencillo temporizador. Situarás dos botones en el formulario, uno para ponerlo en marcha: 'Start' y otro para detenerlo: 'Stop'. Se mostrará en un cuadro de texto el tiempo transcurrido en milisegundos.



Se utilizará un temporizador y el sistema tendrá tres eventos, clic sobre el botón 'Start', clic sobre el botón 'Stop' y el evento del Timer que refrescará el cuadro de texto. Dibujar un esquema de su funcionamiento.

Ajustar su tamaño, color, posición y cambiar el nombre por defecto utilizando el inspector de propiedades.