

Diseño de Aplicaciones con Model Maker for Delphi	1
Creación de un diagrama de clases	1
Agregar una nueva clase al diagrama	2
Agregación de atributos y métodos a la clase	3
Agregar un nuevo atributo	3
Agregar un nuevo método.....	4
Implementación de código en la clase	5
Generación del archivo con código.....	6
Generar el archivo físico de la unidad	8

Diseño de Aplicaciones con Model Maker for Delphi

Al iniciar el programa Model Maker versión 6.2, nos presentará la pantalla que observamos en la figura 1. Esta vista es la de un proyecto nuevo; en caso de ya tener alguno generado, será cargado por default la primera vez.

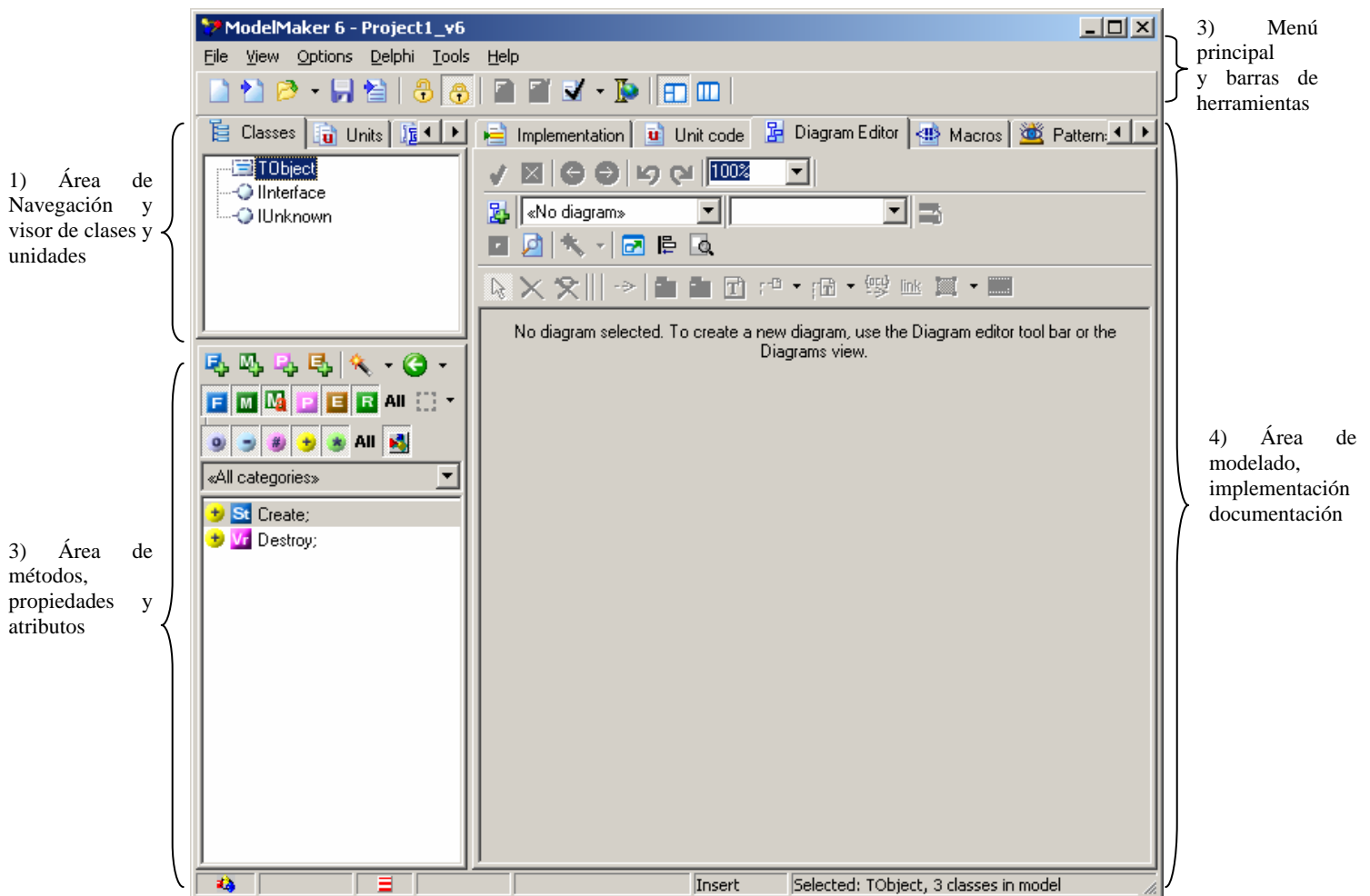


Figura 1. Pantalla Inicial de Model Maker

Creación de un diagrama de clases

En esta sección se describe el proceso para crear una nueva clase con ayuda de la herramienta. El proceso de creación consiste en definir el nombre de la clase, sus atributos y métodos. Una vez hecho esto se procede a la implementación y generación del código de la clase.

Como primer paso iremos dentro del área de navegación a la sección de *diagrams* o bien dando tan solo la tecla F5. La ventana de la figura 2, nos muestra las secciones importantes que se utilizarán en la creación de la clase.

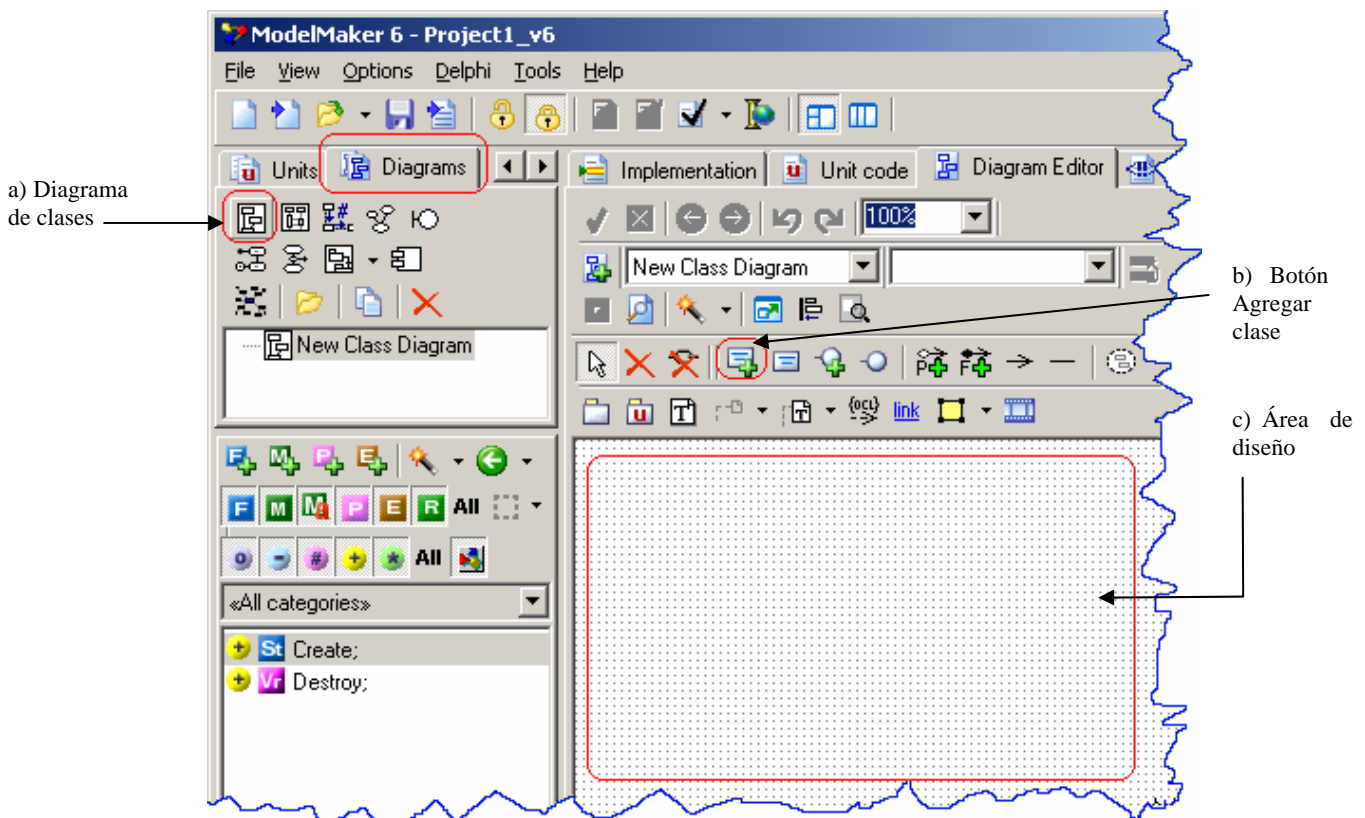


Figura 2. Diseño de diagramas de clase

Agregar una nueva clase al diagrama

1. De clic en el botón marcado por el inciso (a), en la figura 2. Esto agregará un nuevo diagrama llamado *New Class Diagram* y un área de diseño mostrada por el inciso (c) en la misma figura.
2. Ahora de clic en el botón de agregar nueva clase indicado por el inciso (b).
 - a. Ahora de clic en el área de diseño
 - b. Se abre el cuadro de dialogo *class symbol*, que se muestra en la figura 4, en la página siguiente
 - c. Escriba el nombre de la clase deseada, en este ejemplo **Sumar**. Note que todas las clases por default heredan de TObject (ver el combo Ancestor).
 - d. De clic en ok.
3. A partir de esta acción podrá ver sobre el área de diseño una clase en tono amarillo (color por default), tal y como se muestra en la figura 3.

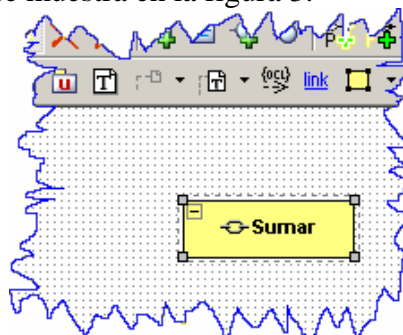


Figura 3. Clase sumar agregada al diagrama de clases

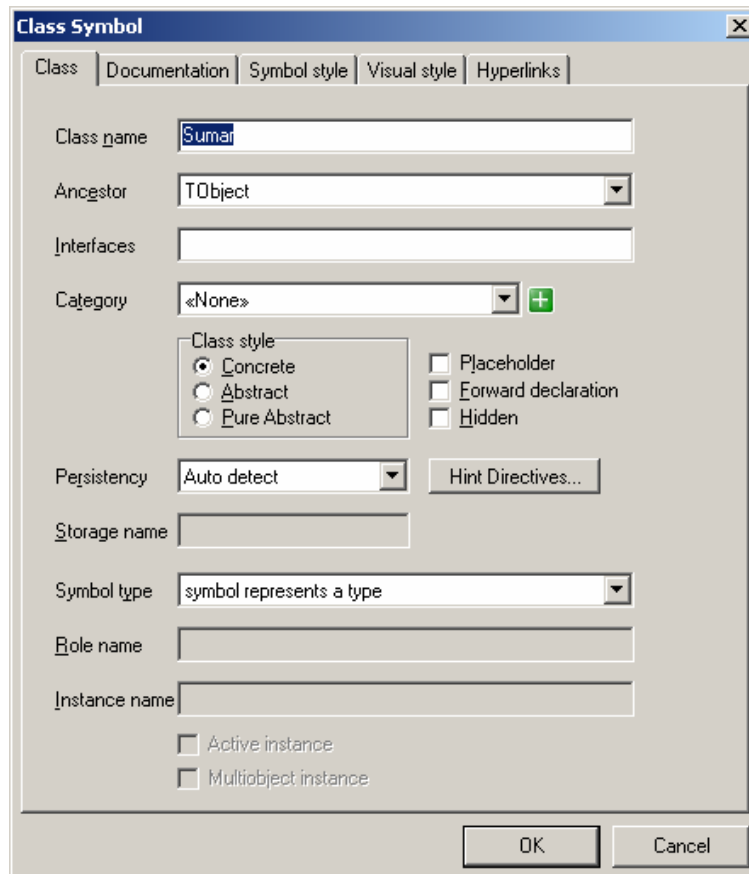


Figura 4. Cuadro de dialogo, símbolo de clase.

Agregación de atributos y métodos a la clase

A nuestra recién creada clase llamada Sumar, procederemos a agregarle atributos y métodos. Para ello deberemos hacer uso de las herramientas que se muestran en la figura 5:

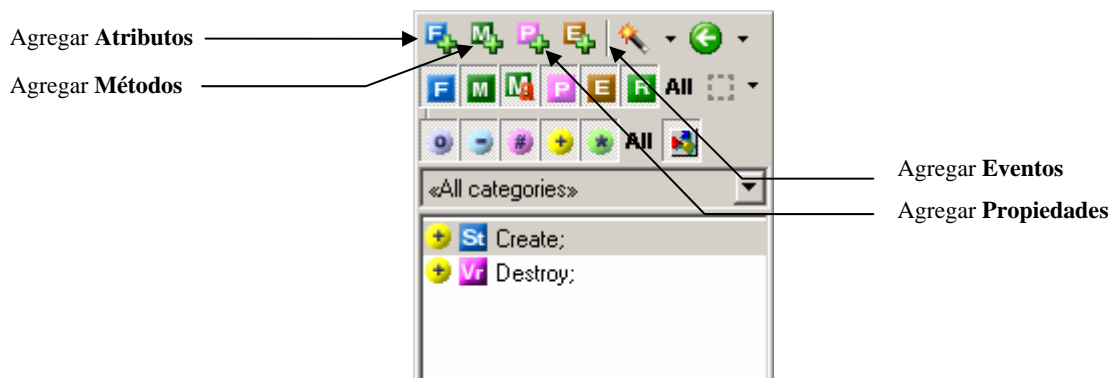



Figura 5. Botones de agregación de métodos y atributos

Agregar un nuevo atributo

1. Seleccionar la clase en el diagrama de diseño
2. De clic en el icono de agregar un nuevo atributo 
 - a. Esta acción desplegará el cuadro de dialogo siguiente:

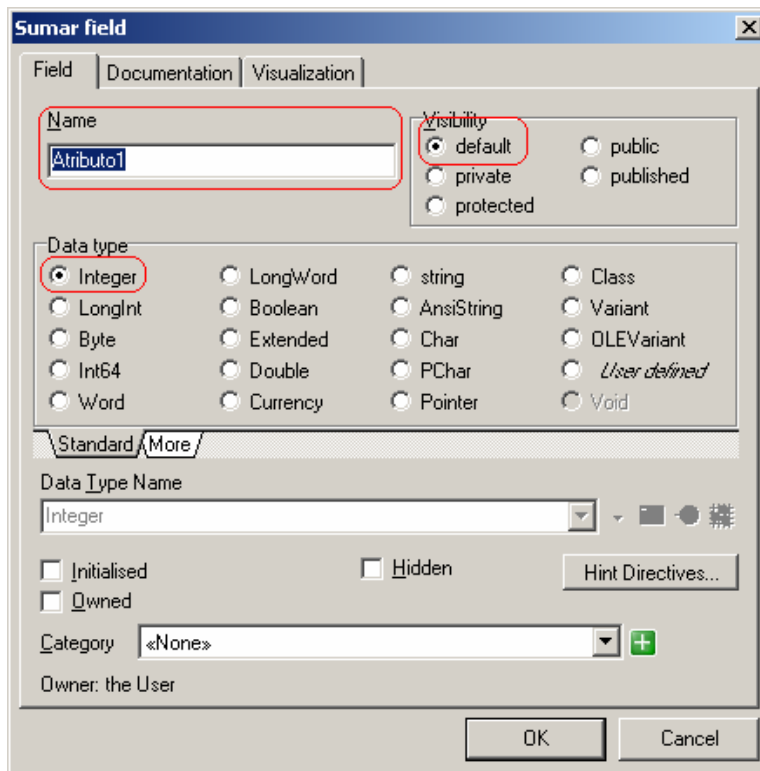



Figura 6. Agregación de atributos

- b. Teclee el nombre del atributo, para este ejemplo *Atributo1*
- c. Establezca la visibilidad, para este ejemplo *default*
- d. Establezca el tipo de dato para el atributo, para este ejemplo *Integer*

Agregar un nuevo método

Un proceso similar debe seguirse para la agregación de métodos y propiedades a la clase, en este caso se describe la agregación de métodos a continuación:

1. Seleccionar la clase en el diseño actual
2. De clic en el icono de agregar nuevo método 
 - a. Esta acción desplegará el cuadro de dialogo agregar método que se puede ver en la figura 7, en la página siguiente.
 - b. En el cuadro de texto *Name* ver figura 7 inciso (a) estableceremos el nombre del método *calcular_suma* es elegido para este ejemplo en particular.
 - c. Al método *calcular_suma*, le vamos a pasar dos parámetros *pr1* y *pr2*, para ello teclearemos en la sección *Parameters* el código apropiado, ver figura 7 inciso (b)
 - d. Finalmente en este ejemplo se decidió que el método devuelva un resultado, por ello el tipo de método será una función, para establecer el tipo de valor retornado seleccionamos la opción apropiada en la sección *Data type*, ver inciso (c) de la misma figura.

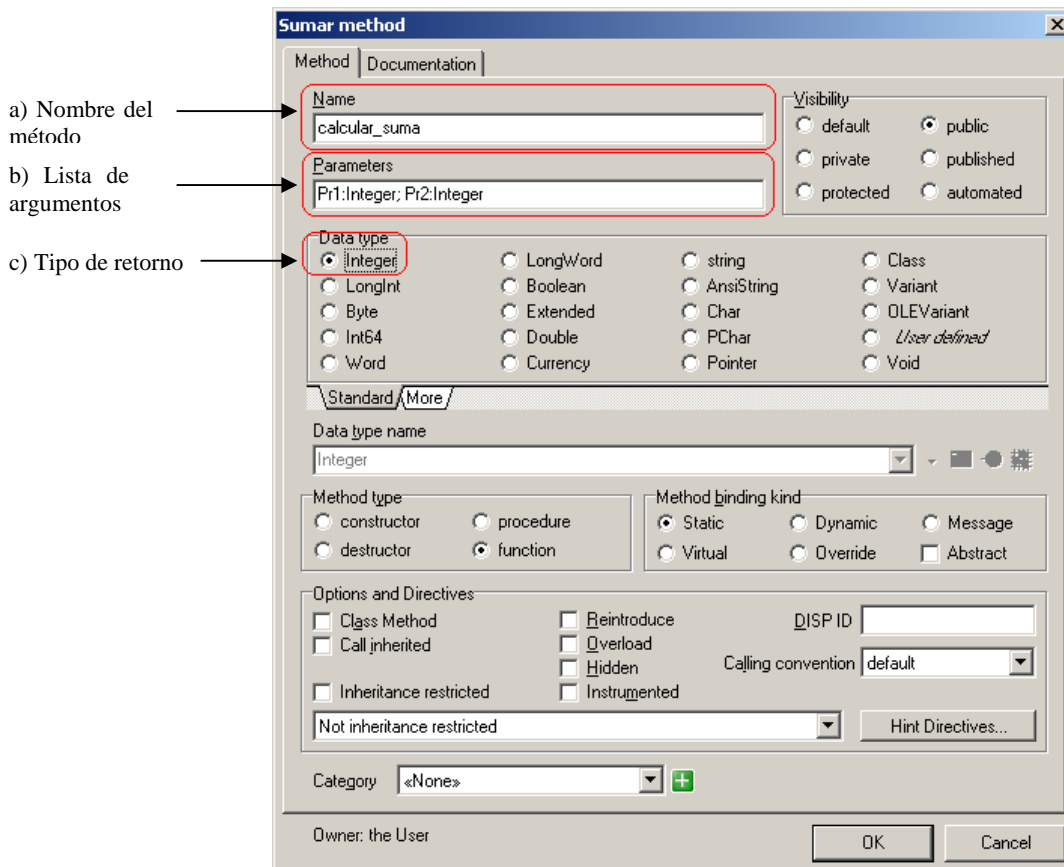


Figura 7. Cuadro de diálogo agregar método

Implementación de código en la clase

Definido el método pasaremos a establecer su implementación. En este caso el objetivo del método `calcular_suma` es el devolver la suma de dos enteros. El código para ello es tan simple como:

```
calcular_suma := pr1 + pr2;
```

Para integrar el código dentro de Model Maker se realizan los pasos siguientes:

1. Seleccionar la carpeta *Implementation* en el área respectiva ver figura 8, inciso (a).
2. En el área de navegación seleccionaremos el nombre del método que deseamos implementar inciso (b). En este caso `calcular_suma`
3. En el área de implementación teclear todo el código apropiado, ver inciso (c).

El ejemplo completo se ve capturado en la figura 8.

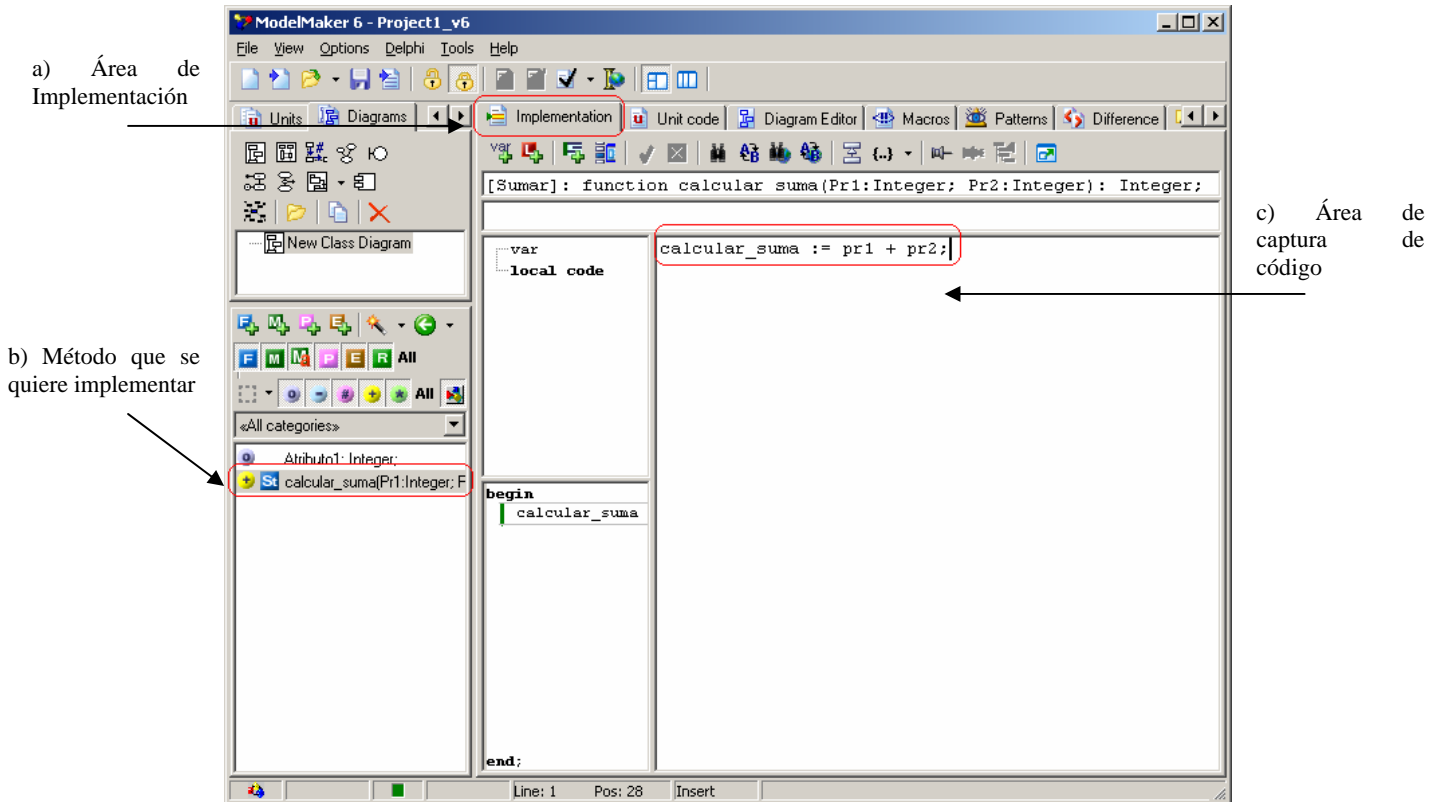




Figura 8, Implementación de código para el método calcular_suma

Generación del archivo con código

Realizado el proceso de definir la clase y escribir el código que la misma tendrá, deberemos decirle a Model Maker que nos genere un archivo de código fuente. A este proceso se le denomina crear unidad.

Para crear la unidad (archivo físico de código) se realizarán los pasos siguientes:

1. Dar clic en el icono de desbloquear generación de código identificado por el icono , esta acción dará como resultado que se habiliten la herramienta de generación de código denotada por el icono 
2. Seleccionar el área de *unit code* en el área de implementación o simplemente teclear F7
3. Seleccionar la carpeta de *Units* en el área de navegación de clases o teclear F4
4. En el área de navegación de unidades
 - a. Seleccionaremos la clase Sumar
 - b. De clic con el botón derecho sobre la clase y seleccione del menú contextual *add unit*. Esta acción abrirá el cuadro de dialogo *Edit Unit*, ver figura 9.
 - c. Seleccione la ruta en donde desea almacenar el archivo y establezca el nombre para el mismo. Cuide que el nombre del archivo sea diferente al de la clase. Para este ejemplo el archivo se llamara **Csumar.pas**, ver inciso (a).
 - d. Seleccione las clases o clase que desea agregar a la unidad de la lista de clases disponibles (*available classes*), para este ejemplo Sumar, ver inciso (b).
5. De clic en ok
6. Como resultado de todos estos pasos se obtiene la figura 10

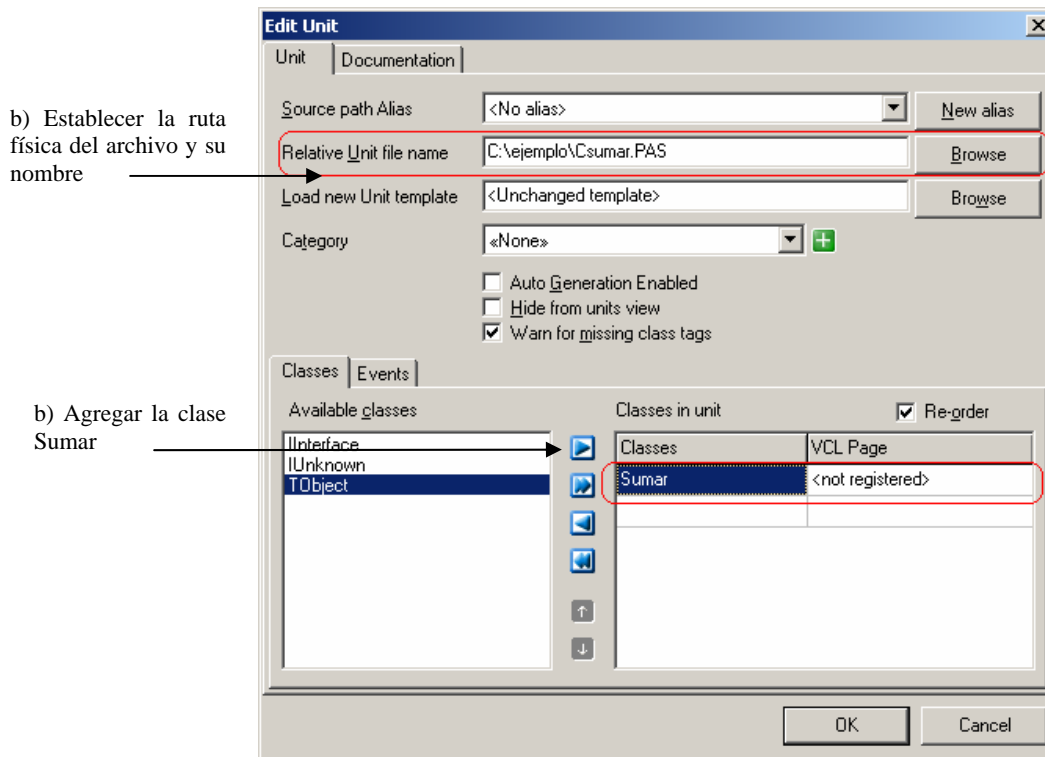


Figura 9. Cuadro de dialogo editar unidad

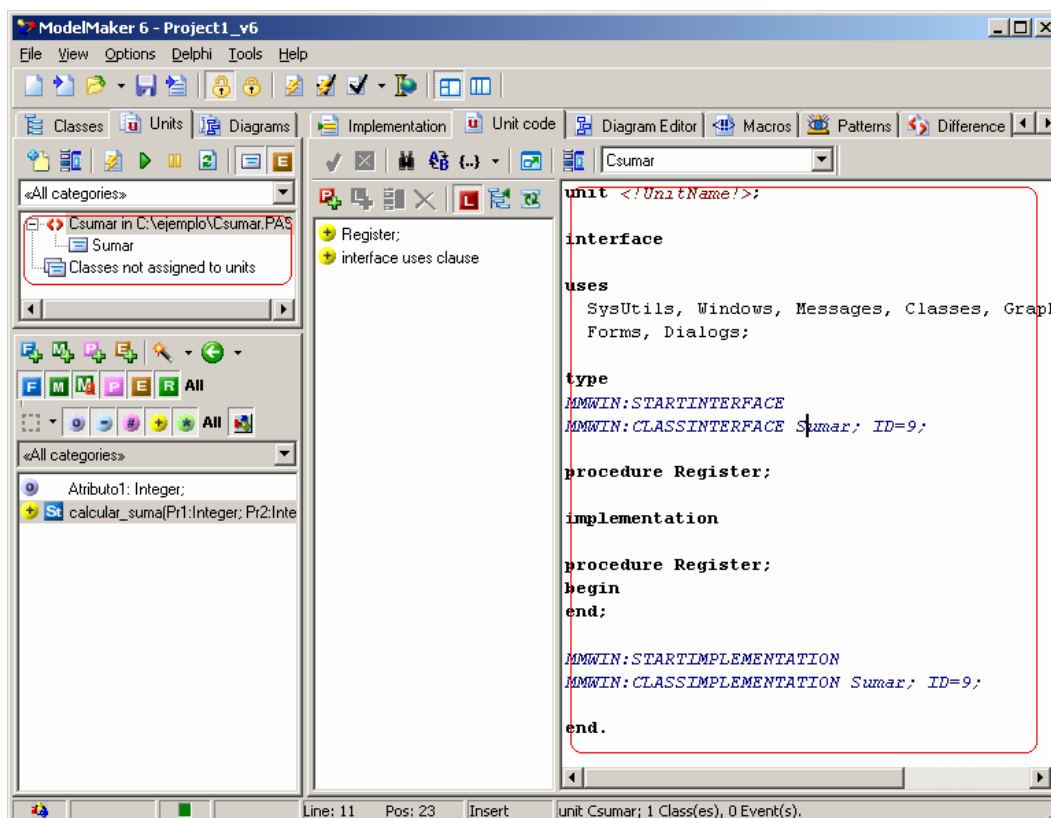



Figura 10. Código para unidad Csumar generado

El último paso es generar físicamente el archivo, ya que los pasos anteriores propiamente tienen como propósito “escribir” el código que forma parte del cuerpo de la unidad. De hecho si observa en la figura 10, notará que hay una sección de código en tono azul, dicho código es una macro que será sustituido por los respectivos atributos y métodos definidos previamente en los apartados anteriores.

Generar el archivo físico de la unidad

1. Seleccionar la clase deseada que se quiere crear
2. Presionar el botón de *generate associated unit* 

Como resultado tendremos en nuestra PC el archivo de salida como se muestra en la figura 11.

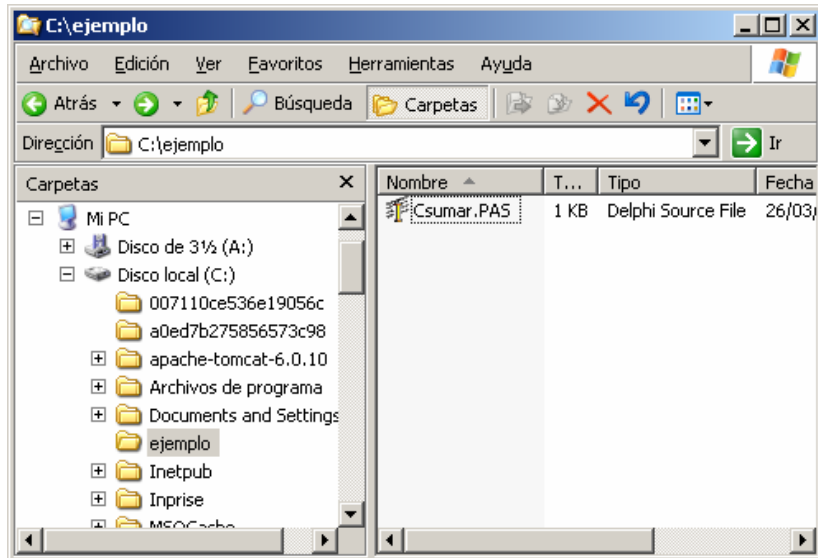


Figura 11. Archivo con código fuente generado

Para conocer como queda el archivo de salida se muestra su código a continuación:

```
unit sumar;  
  
interface  
  
uses  
  SysUtils, Windows, Messages, Classes, Graphics, Controls,  
  Forms, Dialogs;  
  
type  
  suma = class (TObject)  
    param1: Integer;  
    param2: Integer;  
  public  
    function calcular_suma(pr1:Integer; pr2:Integer): Integer;  
  end;  
  
procedure Register;  
  
implementation  
  
procedure Register;  
begin  
end;  
  
{  
***** suma *****  
}  
function suma.calcular_suma(pr1:Integer; pr2:Integer): Integer;  
begin  
  calcular_suma := pr1 + pr2;  
end;  
  
end.
```

Figura 12. Código generado por el Model Maker