Programación con Delphi (I)

© Francisco Charte Ojeda – http://www.fcharte.com

Sumario

Crear sus propias aplicaciones para Windows no es una tarea compleja en exceso. En este curso podrá aprender las bases para hacerlo, usando una de las herramientas más conocidas y potentes: Borland Delphi 5.

Introducción

La mayoría de los usuarios de un ordenador son, a su vez, usuarios de aplicaciones informáticas que se adaptan, más o menos, a sus necesidades. Muchas de estas aplicaciones, como las hojas de cálculo, las bases de datos o los procesadores de textos, son de uso genérico, no aplicaciones hechas a medida para un usuario determinado. Diseñar y crear sus propios programas, lo que tradicionalmente se ha entendido por *programar*, es cada día más fácil, gracias a la existencia de herramientas como Borland Delphi. En éstas, gran parte del trabajo se efectúa de manera visual, arrastrando y soltando objetos con el ratón, en lugar de escribiendo gran cantidad de código.

En la serie que se inicia con esta entrega podrá aprender a crear aplicaciones a medida utilizando Borland Delphi. El punto de partida, esta primera entrega, la dedicaremos a introducir términos y conceptos genéricos, necesarios para poder avanzar posteriormente. Lo único que se asume es que el lector es usuario de Windows, habituado, por tanto, al uso del ratón y con conocimiento de técnicas como arrastrar y soltar.

Lógicamente, para poder seguir los ejemplos propuestos deberá disponer de alguna de las versiones de Delphi instalada en su equipo. Nosotros usaremos la edición *Enterprise* de Borland Delphi 5. La última versión disponible en este momento es la 6. No obstante, las diferencias respecto a otras versiones previas serán mínimas y, siempre que sean de importancia, se indicarán adecuadamente.

El entorno de Delphi

La instalación de Delphi en el sistema es una tarea muy sencilla, similar a la instalación de cualquier otra aplicación, por lo que no emplearemos tiempo en ello y asumiremos que Delphi ya se encuentra instalado. Para iniciar Delphi se procederá, asimismo, como en cualquier otro caso, por regla general abriendo el menú del botón **Inicio** y seleccionando la opción adecuada.

El entorno de Delphi es configurable, adaptable a las preferencias del usuario, de ahí que su aspecto pueda variar mucho según los cambios que se hayan efectuado en él. En la figura 1 puede ver el aspecto típico que muestra Delphi al iniciarse, mientras que en la figura 2 puede ver el mismo entorno tras algunos cambios. A partir de la versión 4, las distintas ventanas de Delphi son acoplables entre sí, formando mosaicos de ventanas adosadas o ventanas con múltiples páginas, según nuestras preferencias. Como en muchos otros casos, lo único que hay que hacer es arrastrar y soltar las ventanas hasta allí donde se desean colocar.

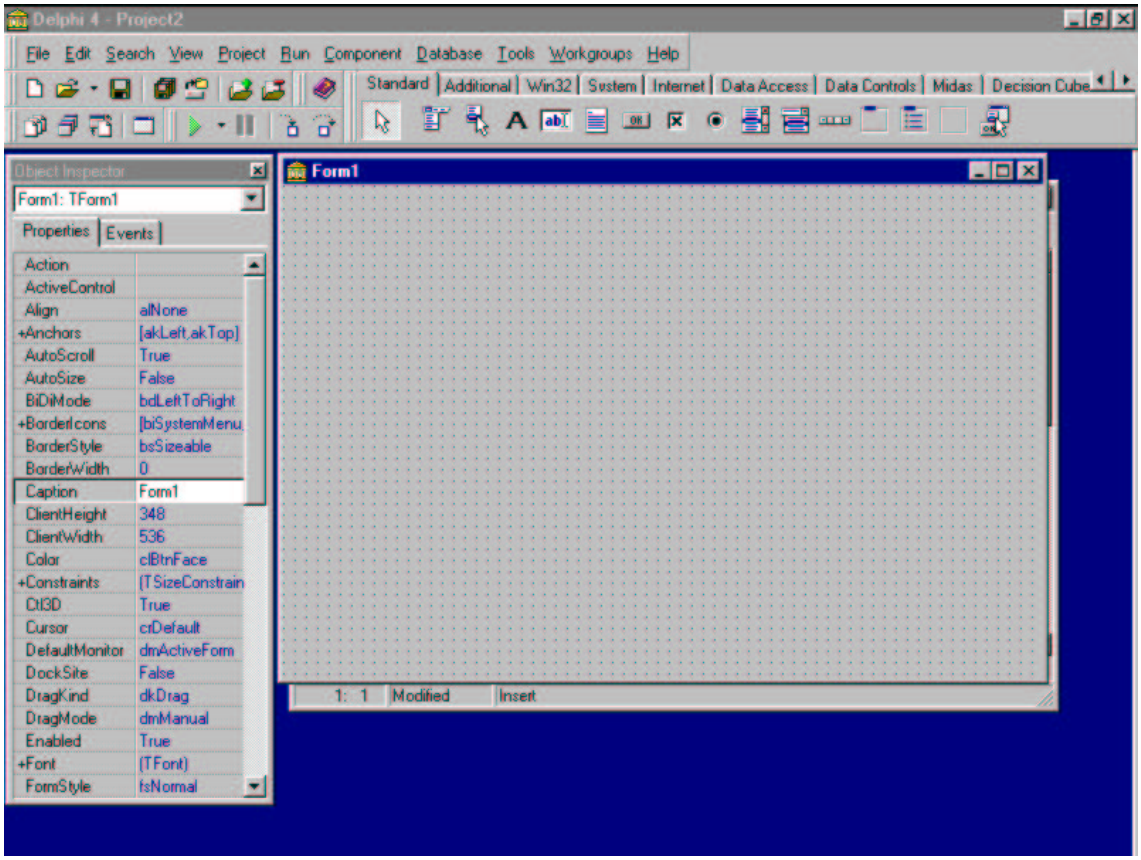

Figura 1. Aspecto por defecto de Delphi 4 al iniciarse, sin ninguna personalización

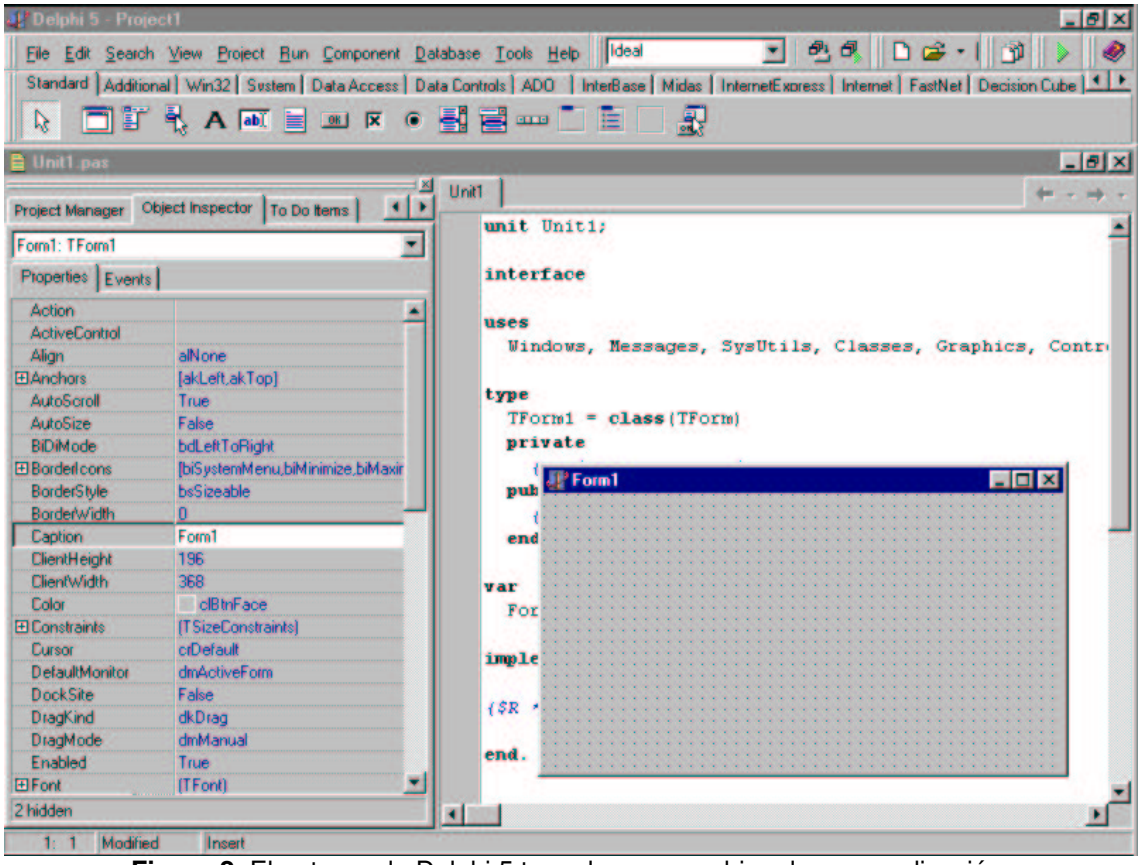

Figura 2. El entorno de Delphi 5 tras algunos cambios de personalización.

Las barras de botones, que puede ver en detalle en la figura 3, ponen a nuestro alcance las acciones más habituales. Con estos botones es posible crear nuevas ventanas de aplicación, ejecutar el proyecto, guardarlo, etc., sin necesidad de recurrir al menú de opciones. Observe que los botones se encuentran agrupados en varias barras, que pueden ocultarse, mostrarse y moverse de manera independiente.

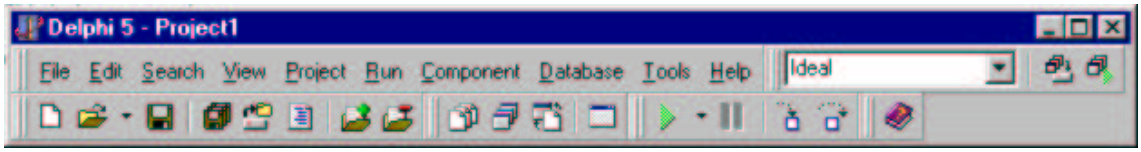

Figura 3. Detalle de la ventana principal de Delphi, con el menú de opciones y las paletas de botones

En la Paleta de componentes (detalle de la figura 4), alojada normalmente junto al menú de opciones y las barras de botones, se encuentran los elementos básicos sobre la base de los cuales se construyen las aplicaciones. Cuando se utiliza un programa de diseño para crear un gráfico, las entidades básicas son las líneas, polígonos, tramas e imágenes. De manera análoga, al usar Delphi para crear una aplicación dichas entidades son los botones, listas, opciones de selección, etc.

Figura 4. La Paleta de componentes cuenta con varias páginas, encontrando en cada una de ellas una serie de elementos conocidos como *componentes*. Éstos son los elementos básicos utilizados para diseñar aplicaciones con Delphi

Si para dibujar se utiliza como soporte un lienzo, que contendrá las diversas entidades citadas, en Delphi se utiliza lo que se conoce como *formulario*. Éste, como puede apreciarse en la figura 5, es una ventana que servirá como contenedor, alojando a todos los componentes que se necesiten para conseguir la funcionalidad que se espera del programa.

Ciertos elementos del entorno de Delphi, como las barras de botones o la Paleta de componentes, no sólo pueden ser colocados donde nos interese, ya sea como ventanas independientes o adosadas a otras, sino que, además, su contenido puede ser también modificado. Usando el botón secundario del ratón, para hacer aparecer el correspondiente menú contextual, podrá tanto añadir como eliminar botones, así como modificar las páginas de la Paleta de componentes y los objetos que hay en ellas.

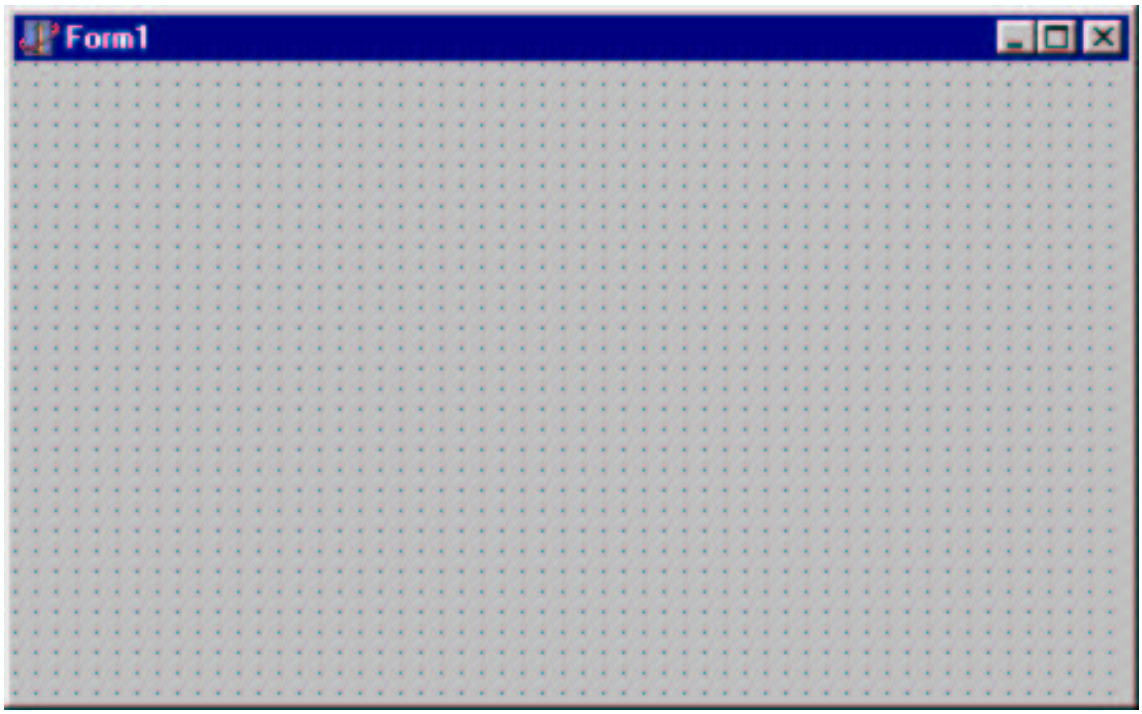

Figura 5. El formulario en una aplicación Delphi es como un lienzo de dibujo en el que podemos insertar diversos elementos, los componentes, que compondrán el programa

Componentes personalizables e interactivos

Cuando se está utilizando un programa de dibujo, por continuar con la analogía usada en el punto anterior, está claro que todos los triángulos son triángulos y, lógicamente, todas las circunferencias son circunferencias. No obstante, un triángulo puede diferenciarse de otro por sus dimensiones, por su tipo (isósceles, escaleno o equilátero), su color, etc. Dicho en otras palabras: un triángulo es siempre un triángulo, pero nosotros podemos personalizarlo durante el dibujo.

Con los elementos usados para crear programas en Delphi, los componentes, ocurre otro tanto. Todos los botones son botones pero, dependiendo de nuestras necesidades, podemos personalizarlos adecuadamente. Con este fin, los componentes cuentan con *propiedades*, como pueden ser las dimensiones, el color, el título que aparece en el interior del botón, etc.

Para modificar las propiedades de los componentes, personalizándolos según necesitemos, Delphi cuenta con una ventana conocida como Inspector de objetos (véase la figura 6). En el Inspector de objetos aparece una lista con todas las propiedades del componente que se elija, mostrándose tanto su nombre como su contenido. El color de un botón, por ejemplo, se almacena en una propiedad llamada `Color`, y sus posibles valores serían `clWhite`, `clRed` o `clYellow`.

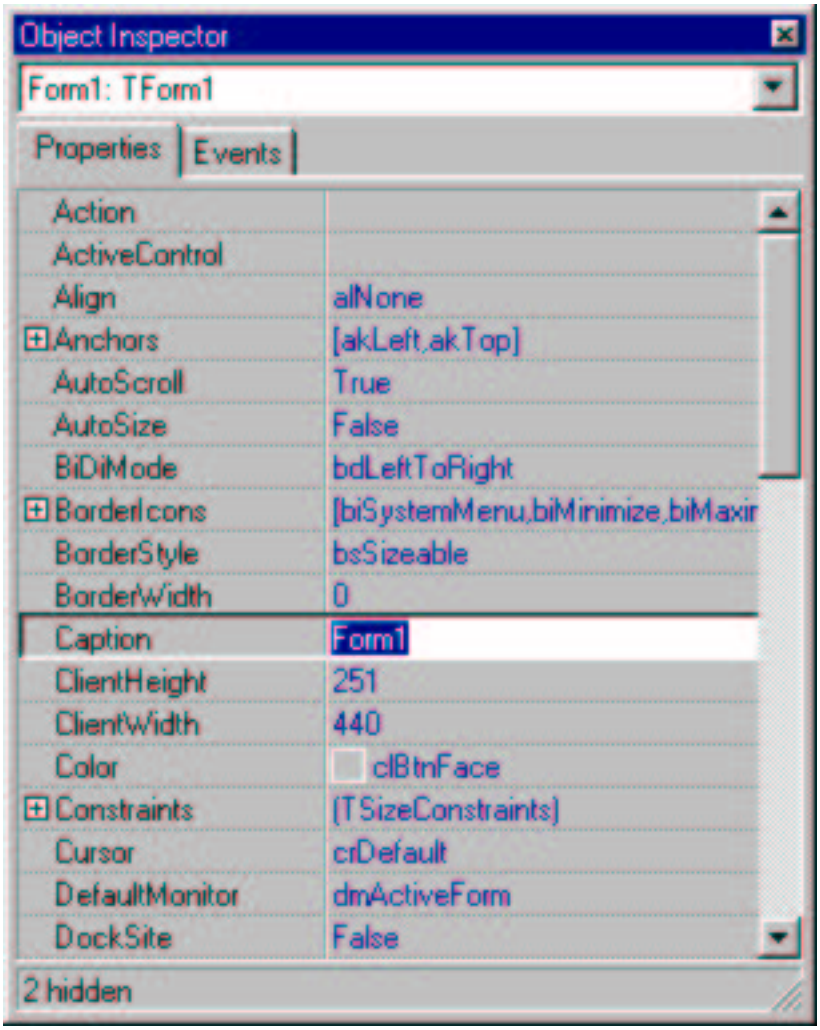

Figura 6. Con el Inspector de objetos es posible modificar las propiedades de los componentes, adaptándolos a nuestras necesidades.

Un dibujo, una vez que se ha terminado, suele ser algo estático. El *espectador* puede verlo, analizarlo y opinar sobre él, pero no puede interactuar con él. Un programa, por el contrario, se construye generalmente para que el usuario final pueda utilizarlo, interactuando con él. Estas interacciones se producen casi siempre a través de dos dispositivos: el teclado y el ratón. Con ellos el usuario puede introducir un nombre, pulsar un botón o seleccionar una opción de un menú. Cada vez que se efectúa una de estas acciones, el sistema operativo la traduce en un mensaje que envía a la aplicación. Estos mensajes son conocidos como *eventos*.

El lenguaje de programación

A pesar de la simplicidad que aporta el desarrollo de programas utilizando componentes, efectuando tareas que hace tiempo debían realizarse mediante la escritura de código, para crear una aplicación no basta con pulsar botones y arrastrar y soltar objetos, también es preciso escribir algo de código. Este código se introduce utilizando un cierto lenguaje de programación que, en el caso de Delphi, es *Object Pascal*.

El lenguaje Pascal es uno de los más conocidos desde su creación, a principios de la década de los setenta. Derivado de ALGOL, Pascal es un lenguaje que ha influido en otros muchos, desde Ada hasta Modula, Oberon o el propio Visual Basic. Su punto álgido, en cuanto a uso se refiere, se produjo en la década de los ochenta, con la aparición del Turbo Pascal de Borland, del cual es heredero el actual Delphi.

Pascal es un lenguaje inicialmente pensado para la educación aunque, hasta la actualidad, ha sido utilizado con muchos otros fines. Desde sus inicios ha sido un lenguaje claro, estructurado y elegante, fácil de aprender y útil en la mayoría de campos de aplicación. Object Pascal es una evolución del Pascal original que, preservando esa claridad y elegancia, añade al lenguaje los elementos necesarios para convertirlo en un lenguaje orientado a objetos, al estilo de C++.

Como se ha indicado anteriormente, Delphi recibe del sistema unos eventos o señales en determinadas circunstancias, por ejemplo cuando se pulsa un botón del ratón o del teclado. Siempre que nos interese, utilizaremos el lenguaje Object Pascal para describir qué deseamos hacer cuando se reciba un evento. Dicho código será introducido en el Editor de código, que puede ver en la figura 7.

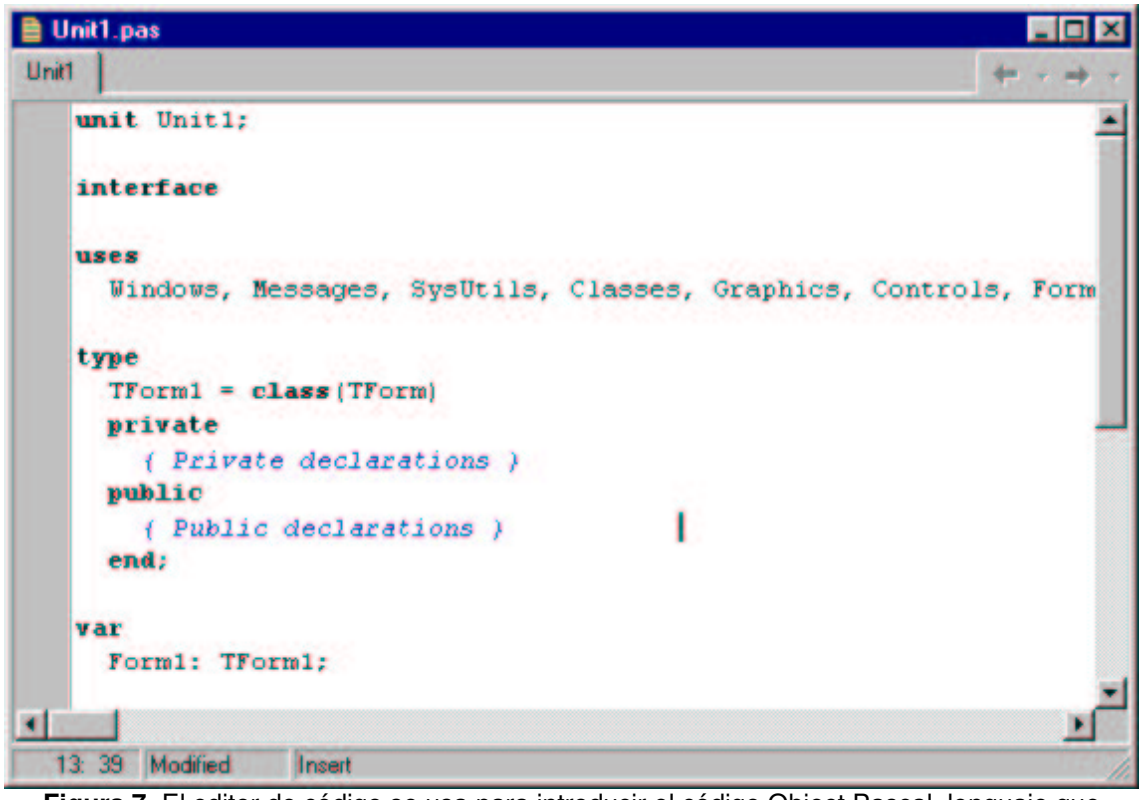

Figura 7. El editor de código se usa para introducir el código Object Pascal, lenguaje que reconoce y del que diferencia mediante colores los distintos elementos sintácticos.

Visto y por ver

Esta primera entrega nos ha servido para familiarizarnos con el entorno de Borland Delphi, conociendo los nombres con los que se denomina a los elementos más importantes: Paleta de componentes, Inspector de objetos, Editor de código, etc. También se ha descrito, a grandes rasgos, el proceso de desarrollo de una aplicación usando Delphi: diseño de la interfaz, insertando componentes en

un formulario, y codificación de la funcionalidad, escribiendo código asociado a los eventos que generan esos componentes.

En la próxima entrega entraremos en algunos detalles acerca del diseño de la interfaz, insertando componentes y utilizando el Inspector de objetos para personalizarlos. También aprenderá a asociar una línea de código a un evento, creando su primer programa funcional que, lógicamente, podrá ejecutar y ver en funcionamiento.

Desarrollo basado en componentes

Durante muchos años, prácticamente desde los inicios de la informática hasta principios de esta década, la programación o desarrollo de aplicaciones para ordenadores ha sido una tarea prácticamente artesanal. El equipo implicado en el desarrollo, muchas veces una sola persona, tenía que contar con conocimientos en muy distintas áreas, desde la gestión de bases de datos hasta las comunicaciones, pasando por los gráficos por citar los campos más conocidos. Utilizando una analogía, sería como que para construir un coche el fabricante tuviese que crearlo todo desde cero, desde las ruedas hasta el motor, pasando por los más mínimos detalles del equipamiento.

Lógicamente, la producción de un bien con esta técnica limita mucho sus posibilidades y alcance. Imagine cuántos de nosotros dispondríamos de un automóvil si no existiese la fabricación en cadena, los fabricantes especializados y las cadenas de montaje.

Actualmente, por fortuna, el desarrollo de software ha cambiado considerablemente gracias a la existencia de los componentes, que no son mas que piezas del puzzle que será la aplicación. Nuestro trabajo, como programadores, consistirá en seleccionar los componentes adecuados, personalizarlos y conectarlos entre sí para conseguir la funcionalidad perseguida. Es como crear una construcción a partir de las piezas de un mecano, más o menos.

Borland Delphi cuenta con un importante número de componentes que, como se indicó previamente, encontrará en las distintas páginas de la Paleta de componentes. A pesar de ello, esos componentes no cubren todas las necesidades de todos los programadores, algo que sería prácticamente imposible. No obstante, existen cientos de desarrolladores que se dedican a crear componentes, objetos que pueden ser añadidos a la Paleta de componentes de Delphi y utilizados en sus aplicaciones directamente, como haría con los componentes que ya incorpora de por sí esta herramienta.