



Embarcadero Delphi, antes conocido como CodeGear Delphi, Inprise Delphi y Borland Delphi, es un entorno de desarrollo de software diseñado para la programación de propósito general con énfasis en la programación visual. En Delphi se utiliza como lenguaje de programación una versión moderna de Pascal llamada Object Pascal. Es producido comercialmente por la empresa estadounidense CodeGear (antes lo desarrollaba Borland), adquirida en mayo de 2008 por Embarcadero Technologies, una empresa del grupo Thoma Cressey Bravo, en una suma que ronda los 30 millones de dólares. En sus diferentes variantes, permite producir archivos ejecutables para Windows, GNU/Linux y la plataforma .NET.

CodeGear ha sido escindida de la empresa Borland, donde Delphi se creó originalmente, tras un proceso que pretendía en principio la venta del departamento de herramientas para desarrollo.

Un uso habitual de Delphi, aunque no el único, es el desarrollo de aplicaciones visuales y de bases de datos cliente-servidor y multicapas. Debido a que es una herramienta de propósito múltiple, se usa también para proyectos de casi cualquier tipo, incluyendo aplicaciones de consola, aplicaciones de web (por ejemplo servicios web, CGI, ISAPI, NSAPI, módulos para Apache), servicios COM y DCOM, y servicios del sistema operativo. Entre las aplicaciones más populares actualmente destaca Skype, un programa de telefonía por IP.

Delphi inicialmente sólo producía ejecutables binarios para Windows: Delphi 1 para Win16 y con Delphi 2 se introdujo Win32.

## El lenguaje de programación

Delphi está basado en una versión de Pascal denominada Object Pascal. Borland en los últimos años defendía que el nombre correcto del lenguaje es también *Delphi*, posiblemente debido a pretensiones de marca, aunque en sus mismos manuales el nombre del lenguaje aparecía como *Object Pascal*, por lo que la comunidad de programadores no ha adoptado mayoritariamente este cambio (supuesta *aclaración*, según Borland). Object Pascal expande las funcionalidades del Pascal estándar:



- Soporte para la programación orientada a objetos también existente desde Turbo Pascal 5.5, pero más evolucionada en cuanto a:
  - *Encapsulación*: declarando partes privadas, protegidas, públicas y publicadas de las clases
  - *Propiedades*: concepto nuevo que luego han adaptado muchos otros lenguajes. Las propiedades permiten usar la sintaxis de asignación para *setters* y *getters* (en Delphi, *setters*: write y *getters*: read).
  - Simplificación de la sintaxis de referencias a clases y punteros.
- Soporte para manejo estructurado de excepciones, mejorando sensiblemente el control de errores de usuario y del sistema.
- Programación activada por eventos (*event-driven*), posible gracias a la técnica de delegación de eventos. Esta técnica permite asignar el método de un objeto para responder a un evento lanzado sobre otro objeto. Fue adoptada por Niklaus Wirth, autor del Pascal Original, e incorporada a otros de sus lenguajes como Component Pascal.
- Implementación de Métodos, los cuales contienen procedimientos y funciones.
- Manejo de la herencia simple, que permite la reutilización de código, ampliamente utilizada en el desarrollo de componentes.
- Una parte importante, que es de resaltarse, la Programación Multihilos.

La potencia del lenguaje de programación Delphi, se basa principalmente en la gestión y administración de Base de Datos, concepto mediante el cual se repotencio para constituirse en una herramienta importante para desarrolladores, que requerían velocidad y manejo de grandes cantidades de datos, basada en un lenguaje de programación extensamente variado Object Pascal.

Esto permitió a los desarrolladores noveles (principiantes) y avanzados contar con una herramienta clara y de fácil manejo, moderna, capaz de ejecutar con gran velocidad, procedimientos intrínsecos, que permiten obtener resultados inmediatos y a bajo costo. Entre sus características principales, destaca lo atractivo del producto final, debido al apoyo de su interfaz gráfica de desarrollo, por lo que se sitúa en la categoría de lenguajes visuales, pero no todo es bondad, una desventaja con los programas no visuales, es la gran cantidad de



memoria que emplea y que su ejecutable resulta demasiado grande en comparación con uno hecho en lenguaje C, pero esta desventaja se compensa con el tiempo de desarrollo de una aplicación; se sacrifica espacio, pero se gana velocidad.

Esta herramienta de desarrollo, permite emular las características de los lenguajes denominados de bajo nivel. Trabaja tanto en modo consola, como en ambiente gráfico windows, actualmente existe una versión para Linux.

## Componentes

Delphi dio una implementación muy buena a la idea del uso de componentes, que son piezas reutilizables de código (clases) que pueden interactuar con el EID en tiempo de diseño y desempeñar una función específica en tiempo de ejecución. Desde un enfoque más específico de la herramienta, se catalogan como componentes todos aquellos objetos que heredan de la clase *TComponent*, donde se implementa la funcionalidad necesaria para interactuar con el entorno de desarrollo, la carga dinámica desde *streams* y la liberación de memoria mediante una jerarquía. Una gran parte de los componentes disponibles para Delphi son *controles* (derivados de *TControl*), que encapsulan los elementos de interacción con el usuario como botones, menús, barras de desplazamiento, etcétera.

Delphi incluye una biblioteca de clases bien diseñada denominada VCL (*Visual Component Library*, Biblioteca de Componentes Visuales) y, en sus versiones 6 y 7, una jerarquía multiplataforma paralela denominada CLX. Ésta también se incluye en Kylix. Estas jerarquías de objetos incluyen componentes visuales y no visuales, tales como los pertenecientes a la categoría de acceso a datos, con los que puede establecerse conexiones de forma nativa o mediante capas intermedias (como ADO, BDE u ODBC) a la mayoría de las bases de datos relacionales existentes en el mercado. La VCL también está disponible para el desarrollo en .NET.

Además de poder utilizar en un programa estos componentes estándar (botones, grillas, conjuntos de datos, etc.), es posible crear nuevos componentes o mejorar los ya existentes, extendiendo la funcionalidad de la herramienta. En Internet existe un gran número de



componentes, tanto gratuitos como comerciales, disponibles para los proyectos a los que no les basten los que vienen ya con la herramienta.

## Eventos

Delphi permite de manera sencilla ejecutar trozos de código en respuesta a acciones o *eventos* (sucesos) que ocurren durante el tiempo que un programa se ejecuta. Por ejemplo, cuando se presiona un botón, la VCL captura la notificación estándar de Windows, y detecta si hay algún método asociado al evento *OnClick* del botón. Si lo hay, manda ejecutar dicho método.

Los eventos pueden generarse debido a la recepción de señales desde elementos de hardware como el ratón o el teclado, o pueden producirse al realizar alguna operación sobre un elemento de la propia aplicación (como abrir un conjunto de datos, que genera los eventos *BeforeOpen/AfterOpen*). La VCL ha demostrado estar bien diseñada y el control que se tiene a través de los eventos de la misma es suficiente para la gran mayoría de aplicaciones.

## Base de datos

Una de las principales características y ventajas de Delphi es su capacidad para desarrollar aplicaciones con conectividad a bases de datos de diferentes fabricantes. El programador de Delphi cuenta con una gran cantidad de componentes para realizar la conexión, manipulación, presentación y captura de los datos, algunos de ellos liberados bajo licencias de código abierto o gratuitos. Estos componentes de acceso a datos pueden enlazarse a una gran variedad de controles visuales, aprovechando las características del lenguaje orientado a objetos, gracias al polimorfismo.

En la paleta de componentes pueden encontrarse varias pestañas para realizar una conexión a bases de datos usando diferentes capas o motores de conexión.

Hay motores que permiten conectarse a bases de datos de diferentes fabricantes tales como



BDE, DBExpress o ADO, que cuentan con manejadores para los formatos más extendidos.

También hay componentes de conexión directa para un buen número de bases de datos específicas: Firebird, Interbase, Oracle, etcétera.

A continuación un breve resumen (aún recopilándose) de las capas de conexión disponibles para las bases de datos más populares:

- Interbase/Firebird: IBX (InterBase eXpress), IBO (IB Objects), MDO (Mercury Data Objects), \*DBExpress, BDE, FibPlus, Zeos
- Oracle: DOA (Direct Oracle Access), NCOci8
- dBase: BDE
- FoxPro: BDE
- Paradox: BDE
- Microsoft SQL Server: BDE, ADO, \*DBExpress
- mySQL: Zeos (nativo), \*DBExpress, BDE y ADO (usando ODBC)
- PostgreSQL: BDE, ADO, ZEOSDBO

## **Borland Database Engine (BDE)**

Es un motor de conexión a bases de datos de uso bastante amplio y que permite manejar bases de datos de escritorio como dBase, Foxpro y Paradox, además de ofrecer la capacidad para conectarse a servidores SQL locales y remotos. Su uso, va siendo cada vez menor, debido a la pobre gestión de memoria que realiza, sustituyéndolo por componentes más actualizados y especializados como DOAC (Direct Oracle Access Components) , DBExpress o ZEOS, esto sumado a la fiabilidad que están presentando los nuevos gestores de Datos en especial tecnologías como RDO y ADO; los cuales son mantenidos por sus fabricantes, forzando la compatibilidad con las versiones preliminares; liberando al programador de actualizaciones en cuanto a gestión de datos. Actualmente ya no es desarrollado por Codegear.



## Desarrollo visual

Como entorno visual, la programación en Delphi consiste en diseñar los formularios que componen al programa colocando todos sus controles (botones, etiquetas, campos de texto, etc.) en las posiciones deseadas, normalmente usando un ratón. Luego se asocia código a los eventos de dichos controles y también se pueden crear módulos de datos, que regularmente contienen los componentes de acceso a datos y las reglas de negocio de una aplicación.

## Entorno Integrado de Desarrollo (EID)

O IDE en inglés (*Integrated Development Environment*), es el ambiente de desarrollo de programas de Delphi. Se trata de un editor de formularios (que permite el desarrollo visual), un potente editor de textos que resalta la sintaxis del código fuente, la paleta de componentes y el depurador integrado, además de una barra de botones y un menú que nos permite la configuración de la herramienta y la gestión de proyectos. En las ediciones *Client/Server* y *Enterprise* el EID también ofrece integración con una herramienta de control de versiones (PVCS).

## Depurador integrado

Es una potente característica que nos permite establecer puntos de ruptura (breakpoints), la ejecución paso a paso de un programa, el seguimiento de los valores de las variables y de la pila de ejecución, así como la evaluación de expresiones con datos de la ejecución del programa. Con su uso, un programador experimentado puede detectar y resolver errores lógicos en el funcionamiento de un aplicativo desarrollado con Delphi. En las ediciones *Client/Server* y *Enterprise* se añade la opción de depuración de programas corriendo en equipos remotos (*remote debugging*), lo que posibilita el uso de todas las características del depurador con un programa ejecutándose en su entorno normal de trabajo y no en el ordenador del programador (en donde muchas veces no ocurren los errores).



## Proyectos alternativos

Existen desarrollos de la comunidad del software libre que intentan producir una versión multiplataforma y completamente independiente de Delphi. La más desarrollada es el llamado Lazarus.

## El nombre *Delphi*

El nombre *Delphi* hace referencia al oráculo de Delfos. Borland eligió ese nombre para resaltar su principal mejora con respecto a su antecesor (Turbo Pascal), que sería su conectividad con bases de datos Oracle (*oráculo*, en inglés). El nombre se suele pronunciar *delfi* en Europa, incluida Gran Bretaña. Se usa *delfái* en inglés de Estados Unidos (existe una población *Delphi* con esa pronunciación), por lo que es la preferida por Borland.

## Turbo Delphi

En septiembre de 2006 Borland lanzó *TurboDelphi* como una versión reducida de Delphi. Hay dos versiones: Turbo Delphi for Windows (*Win32*) y Turbo Delphi for .NET, ambas disponibles en dos ediciones Professional y Explorer, ésta última era gratuita.