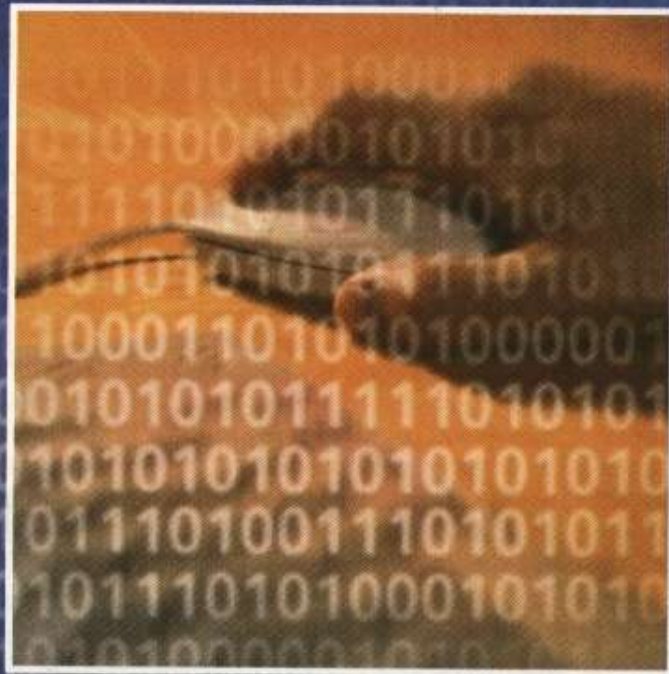


CURSO IBM

PROGRAMAR

es fácil



**Introducción a
la programación**

Multimedia Ediciones, S.A.

Introducción a la programación

AYER Y HOY EN LA PROGRAMACIÓN

Ya no es suficiente con los procedimientos informáticos más básicos que ofrecen las bases de datos o las hojas de cálculo; la exigencia de cálculos más complejos, proyecciones e inferencias, Internet, etc. es cada vez mayor, lo que obliga a la incorporación de técnicas mucho más depuradas, asociadas con la programación de aplicaciones.

La posibilidad de programar una máquina y adaptarla a las necesidades de cada momento y de cada usuario han hecho que los ordenadores se conviertan en máquinas casi indispensables en los entornos más variados, incluyendo los domésticos. Este progreso no habría sido posible sin los avances de la programación y, más concretamente, de los lenguajes y de las herramientas para la automatización de la programación. Los primeros ordenadores se programaban y aceptaban los

datos que se deseaban procesar mediante la modificación de sus circuitos (por ejemplo, sus conmutadores), lo que hacía que el proceso fuese extremadamente lento y costoso.

Más adelante, aunque la forma de programar no varió, los datos comenzaron a ser muy numerosos y no hubo más remedio que introducirlos utilizando cintas y tarjetas perforadas.

El siguiente paso lógico consistió en tratar los programas como datos, almacenándolos en cintas o tarjetas perfo-

radas, con lo que se ganó en versatilidad. Este considerable avance permitió que los programas se pudieran cargar con facilidad y rapidez en la memoria del ordenador.

A pesar de este enorme avance surgieron otras necesidades, ya que entonces fue necesario encontrar una forma fácil y sencilla para comunicarse con el ordenador y codificar los programas que se deseaba ejecutar en la máquina. Con ello aparecieron los primeros lenguajes de programación.

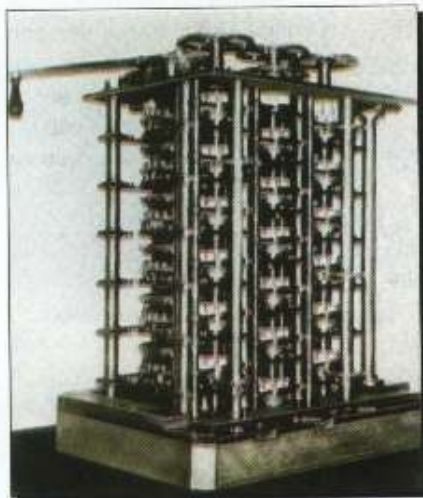


FIG. 1 La máquina analítica de Charles Babbage, creada en 1833, se considera por su estructura el predecesor de los modernos ordenadores.



FIG. 2 Los primeros ordenadores son los bisabuelos de los modernos procesadores que integran millones de transistores en un cuadrado de poco más de 2 cm de lado.

Introducción a la programación

AYER Y HOY EN LA PROGRAMACIÓN

Ya no es suficiente con los procedimientos informáticos más básicos que ofrecen las bases de datos o las hojas de cálculo; la exigencia de cálculos más complejos, proyecciones e inferencias, Internet, etc. es cada vez mayor, lo que obliga a la incorporación de técnicas mucho más depuradas, asociadas con la programación de aplicaciones.

La posibilidad de programar una máquina y adaptarla a las necesidades de cada momento y de cada usuario han hecho que los ordenadores se conviertan en máquinas casi indispensables en los entornos más variados, incluyendo los domésticos. Este progreso no habría sido posible sin los avances de la programación y, más concretamente, de los lenguajes y de las herramientas para la automatización de la programación. Los primeros ordenadores se programaban y aceptaban los

datos que se deseaban procesar mediante la modificación de sus circuitos (por ejemplo, sus conmutadores), lo que hacía que el proceso fuese extremadamente lento y costoso.

Más adelante, aunque la forma de programar no varió, los datos comenzaron a ser muy numerosos y no hubo más remedio que introducirlos utilizando cintas y tarjetas perforadas.

El siguiente paso lógico consistió en tratar los programas como datos, almacenándolos en cintas o tarjetas perfo-

radas, con lo que se ganó en versatilidad. Este considerable avance permitió que los programas se pudieran cargar con facilidad y rapidez en la memoria del ordenador.

A pesar de este enorme avance surgieron otras necesidades, ya que entonces fue necesario encontrar una forma fácil y sencilla para comunicarse con el ordenador y codificar los programas que se deseaba ejecutar en la máquina. Con ello aparecieron los primeros lenguajes de programación.

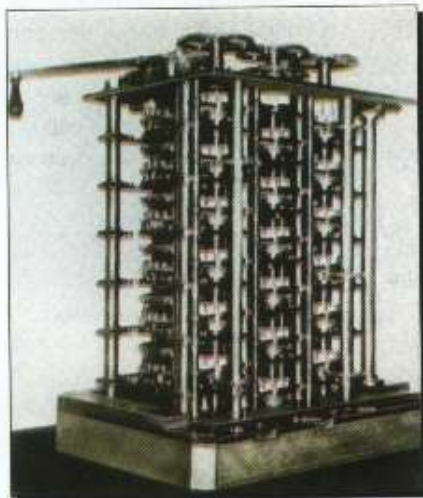


FIG. 1 La máquina analítica de Charles Babbage, creada en 1833, se considera por su estructura el predecesor de los modernos ordenadores.



FIG. 2 Los primeros ordenadores son los bisabuelos de los modernos procesadores que integran millones de transistores en un cuadrado de poco más de 2 cm de lado.



FIG. 3 El lenguaje de programación Pascal tomó su nombre de este ilustre pensador francés.

INTRODUCCIÓN A LA PROGRAMACIÓN DE ORDENADORES

Para que un ordenador lleve a cabo las tareas que le encomendamos es necesario que le indiquemos lo que debe hacer, pero esto no es sencillo, ya que debemos utilizar un lenguaje que la máquina sea capaz de entender.

Esta comunicación con el ordenador se lleva a cabo mediante los recursos que ofrecen los llamados lenguajes de programación.

Todos los lenguajes de programación utilizan un "idioma" con unas re-

glas fijas y convenidas que, en mayor o menor medida, intentan acercar el lenguaje humano al de la máquina, haciendo de intermediarios entre el complicado código máquina y el lenguaje natural.

Debido a que el ordenador sólo entiende y maneja el lenguaje binario formado por dos estados diferentes, representados mediante el 0 y el 1, el conjunto de instrucciones (programa) tiene que sufrir con posterioridad un proceso de transformación, que logre que las instrucciones anteriores sean comprensibles por la máquina.

Por fortuna, la traducción de estos códigos también se presta a su ejecución por el ordenador y tiene la forma de un programa especializado en la traducción.

En la actualidad se emplean dos tipos de programas especiales para realizar esta conversión desde el lenguaje de programación al de la máquina: los traductores y los intérpretes.

El compilador, un traductor al fin y al cabo, acepta programas escritos en un lenguaje (Pascal, C++, C, etc.) y los traduce a otro que sea comprensible por el ordenador en el que se piensa ejecutar, generando un programa equivalente independiente, el



FIG. 4 Normalmente el código de un lenguaje se compila para una determinada arquitectura de ordenador.

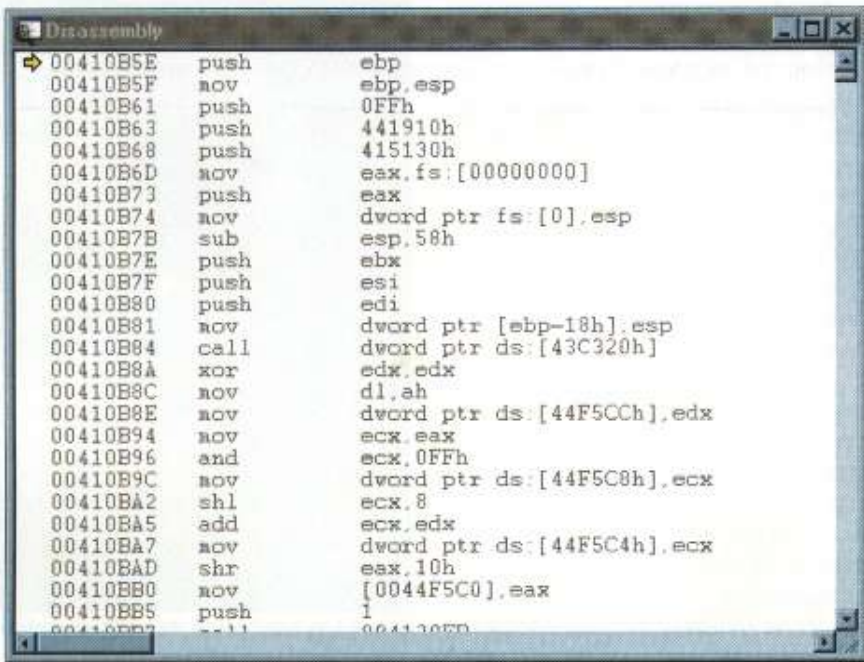


FIG. 5 La introducción del lenguaje ensamblador mejoró en parte la legibilidad de los programas.

cual puede ejecutarse tantas veces como se quiera. En un compilador hay que distinguir tres lenguajes diferentes: el de los programas de partida escritos en un lenguaje más o menos cercano al hombre (código fuente), el de los programas equivalentes traducidos al lenguaje de la máquina y que son específicos para el ordenador en el que se piensa ejecutar el programa (código objeto) y el lenguaje en que está escrito el propio compilador, que puede ser igual o diferente de los otros dos mencionados anteriormente.

Por su parte, los intérpretes aceptan programas escritos en un lenguaje de alto nivel, que puede ser el mismo que se utilizó con el compilador, los analizan y los ejecutan, instrucción a instrucción, bajo el control del propio intérprete. Esto significa que, a diferencia de los compiladores, donde todo el programa se traduce de una vez, el intérprete va traduciendo las instruc-

ciones del programa justo antes de ejecutarse; el intérprete suministrará el código ejecutable al procesador a medida que lo vaya necesitando. Es por ello que la ejecución de los programas interpretados, como los que se generan usando los lenguajes de programación Basic o Java, suele ser más lenta que la de los compilados, como C++ o Pascal.

HISTORIA DE LA PROGRAMACIÓN

Los primeros lenguajes de programación intentaban aprovechar al máximo la pequeña capacidad de los procesadores o las escasas y complejas utilidades que ofrecía su sistema operativo. Así, era frecuente la programación eficiente pero también más compleja, que se realizaba utilizando un lenguaje de bajo nivel denominado

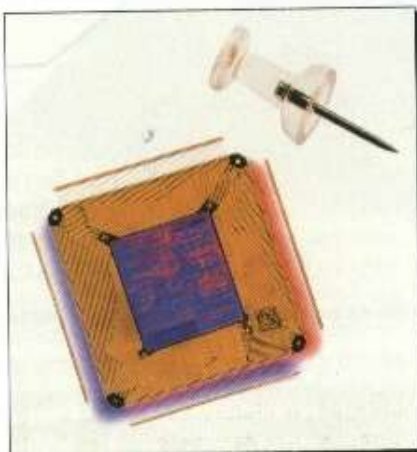


FIG. 6 El avance de la programación ha corrido parejo al del hardware de los ordenadores.



FIG. 7 El incremento de la potencia del hardware dio pie a la aparición de los lenguajes de alto nivel.

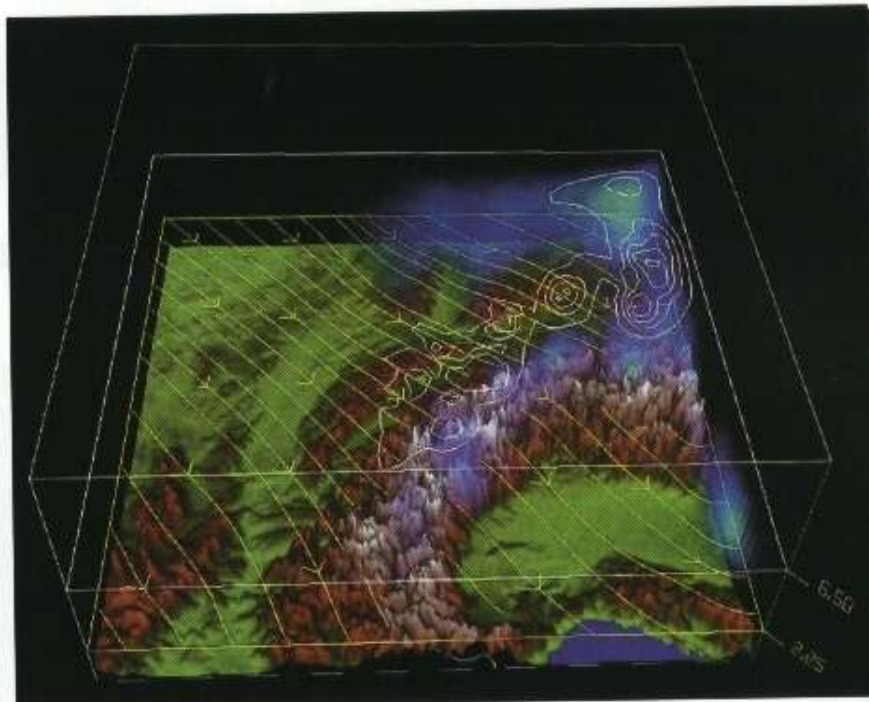


FIG. 8 El diseño de mejores lenguajes, junto con el incremento de las prestaciones de los ordenadores, ha proporcionado a la ciencia una herramienta de estudio de incalculable valor.

ensamblador, que aunque es un lenguaje simbólico, depende mucho de la máquina en la que se desea ejecutar el programa. No cabe duda de que la introducción del ensamblador sirvió principalmente para mejorar la legibilidad de los programas, aunque el código simbólico del ensamblador también introduce instrucciones adicionales, que no corresponden a ninguna instrucción de la máquina. Éstas proporcionan información al compilador y se llaman "pseudoinstrucciones". Por ejemplo, son pseudoinstrucciones las que se encargan de definir el principio y el final de ciertas secciones de los programas (segmentos y procedimientos), o las que dan instrucciones al compilador para que realice su trabajo de una forma u otra.

Pronto se comprobó que los lenguajes simbólicos no eran una solución definitiva para el problema de la programación, ya que si se cambiaba de procesador también era necesario aprender un nuevo ensamblador adaptado a las instrucciones de la nueva máquina: para programar en ensamblador es necesario conocer a fondo el microprocesador, los registros de trabajo de que dispone, la estructura de la memoria y muchas cosas más. A

pesar de sus inconvenientes, los lenguajes simbólicos siguen utilizándose en la actualidad. Algunas ofertas de trabajo exigen a los candidatos conocimientos de ensamblador. Esto se debe, por una parte, a que existe un enorme volumen de código escrito en lengua-



FIG. 9 Los lenguajes de programación son la herramienta de comunicación imprescindible entre el hombre y la máquina.



FIG. 10 Internet y, más concretamente, la World Wide Web deben agradecer a HTML y Java su rápida popularización.

jes de este tipo, del que no puede prescindirse así como así y, por otra, a que todavía se utiliza para programar ciertas secciones críticas de los pro-

gramas, dado que este lenguaje simbólico, al estar muy próximo al nivel de la máquina, permite crear programas especialmente adaptados a ésta, que ocupan un espacio mínimo y funcionan a la máxima velocidad, pues se suele tratar de código muy eficaz.

El siguiente paso lo supusieron los lenguajes de alto nivel, que eran más independientes de la máquina, por lo que cualquier usuario familiarizado con uno de ellos era capaz de programar cualquier ordenador que dispusiera de un compilador o un intérprete para este lenguaje. Estos lenguajes de alto nivel suelen trabajar en dos fases. En la primera traducen el código fuente a un código intermedio, que también es independiente de la máquina. En la segunda traducen dicho código intermedio al lenguaje específico de la máqui-



FIG. 11 Delphi, de la compañía Borland, es un claro ejemplo de lenguaje de programación visual.

na en la que se desea ejecutar el programa. De esta forma, modificando solamente el programa que se encarga de la segunda fase, se puede lograr que un programa pueda traducirse al lenguaje de máquinas distintas con muy poco esfuerzo.

LAS DISTINTAS GENERACIONES DE LOS LENGUAJES DE PROGRAMACIÓN

Los lenguajes de programación se pueden dividir en cuatro generaciones bien diferenciadas, aunque hay autores que proponen cinco. Pese a existir cierto paralelismo, estas generaciones no coincidieron cronológicamente de forma exacta con las del hardware, pero sí aproximadamente, y son las siguientes:

PRIMERA GENERACIÓN

Como hemos dicho, los primeros ordenadores que aceptaron programación se programaban directamente en código binario, que puede representarse mediante secuencias de ceros y unos. Cada modelo de ordenador tenía su propio código, que por esa razón se llama "lenguaje máquina".

SEGUNDA GENERACIÓN

Como trabajar con ceros y unos no resultaba muy cómodo para los programadores, a dichas secuencias se le asociaron códigos mnemotécnicos, más fáciles de entender por los humanos. Así aparecieron los "lenguajes simbólicos" o ensambladores. Estos lenguajes son una mera traducción a símbolos del lenguaje máquina y dependen de la máquina en la que se están utilizando, pero simplifican la escritura de las instrucciones y las hacen más legibles.

TERCERA GENERACIÓN

El siguiente paso dentro de los lenguajes de programación fue la aparición de lo que se conoce como "lenguajes de alto nivel". Estos sustituyen las instrucciones simbólicas del ensamblador por códigos independientes de la máquina, más parecidos al lenguaje humano o al de las matemáticas.

CUARTA GENERACIÓN

La cuarta generación de los lenguajes de programación corresponde a una serie de herramientas, que permiten construir aplicaciones sencillas combinando segmentos de programa prefabricados. En la actualidad hay muchos expertos que piensan que estas herramientas no son, propiamente, lenguajes de programación, pues en muchas ocasiones no son más que extensiones de lenguajes ya existentes. Algunos proponen reservar el nombre de "cuarta generación" para la programación orientada a objetos u OOP (en inglés, Object Oriented Programming).

QUINTA GENERACIÓN

En ocasiones se llama "lenguajes de quinta generación" a los utilizados para crear programas que emplean inteligencia artificial (IA), aunque esta designación ha caído en desuso.

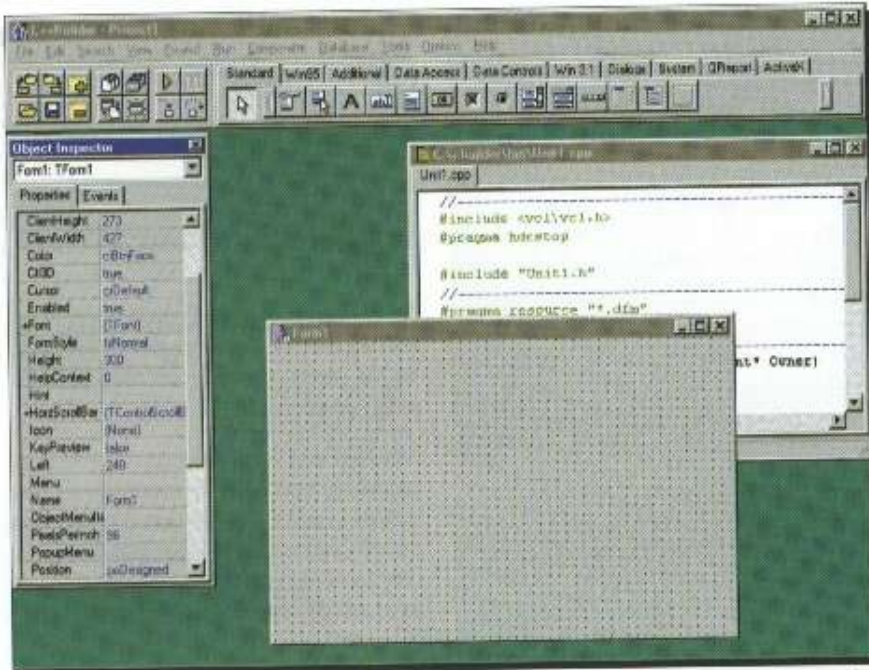


FIG. 12 Mientras que unos consideran a los lenguajes visuales como de cuarta generación, otros proponen reservar esta categoría a la programación orientada a objetos.

Los primeros lenguajes de alto nivel fueron Cobol, Fortran y APL, usados en las décadas de los años 50 y 60. Aunque estos lenguajes simplificaban de forma notable las tareas de programación, estaban prácticamente restringidos al sector de gestión empresarial. Como hemos indicado con anterioridad, todos los lenguajes de programación, e incluso el propio ensamblador, precisan de un proceso de traducción desde su sintaxis nativa hasta el lenguaje que entiende el ordenador (que solamente es capaz de manejar series de unos y ceros). Aunque en la actualidad hay lenguajes que disponen de compiladores e intérpretes, como Basic, hay otros como APL que han sido interpretados, o como Cobol o Fortran,

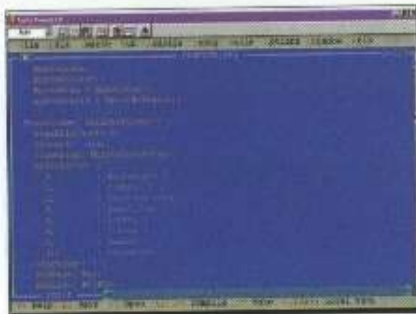


FIG. 13 Pascal es sin duda el lenguaje de programación estructurada por excelencia.

que han sido lenguajes casi exclusivamente compilados. A medida que aparecían en el mercado los distintos lenguajes de alto nivel, los usuarios comprobaron que algunos eran más adecuados que otros dependiendo del tipo de tareas a realizar. Así, Cobol se convirtió en el preferido para realizar aplicaciones de gestión, mientras que For-

tran se impuso en los campos técnicos y de investigación por su facilidad a la hora de manejar fórmulas y cálculos complejos.

Actualmente, y ya más en el entorno del PC, los profesionales de la programación usan habitualmente lenguajes de alto nivel más elaborados y que facilitan mucho más las tareas de programación, como Pascal o C, que en la actualidad ha sido prácticamente desplazado por su variante C++, mucho más eficiente y potente.

Otro lenguaje que ha entrado con muchísima fuerza es Java, que está especialmente pensado para su utilización en Internet, pero con el que se pueden crear aplicaciones de cualquier tipo. Este lenguaje tiene como característica principal su portabilidad entre distintas máquinas y sistemas operativos, y la fácil lectura de su código fuente. Otro lenguaje de programación, aunque con menos posibilidades que el anterior y que también se utiliza en Internet, es HTML (en inglés, *HyperText Markup Language*) y tiene como finalidad la creación de páginas Web. No obstante, ante la necesidad de crear programas complejos de forma rápida y sencilla, aparecieron en el mercado los lenguajes de programación visual, que sobre la base de lenguajes de programación de alto nivel como BASIC,

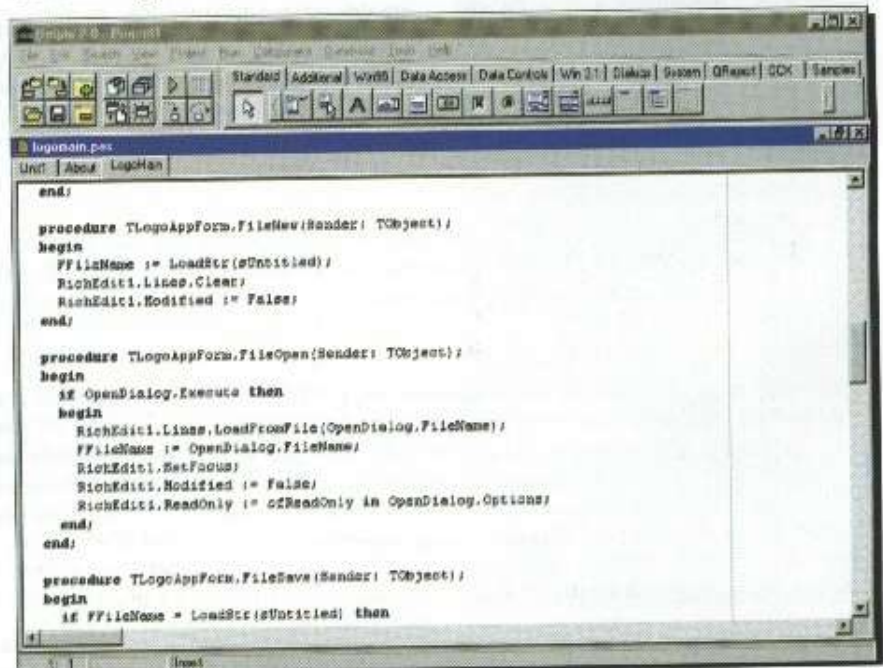


FIG. 14 Algunos lenguajes, como Pascal, han ido evolucionando para adaptarse a los nuevos tiempos.



FIG. 15 La necesidad de ahorrar espacio de almacenamiento fue la causa principal del llamado "bug de fin de milenio".

Pascal, C++ o Java, permiten crear el programa a partir de la selección de los objetos y elementos que se desea que aparezcan en la pantalla. La principal ventaja de los lenguajes visuales es que los usuarios pueden utilizar sus entornos gráficos de desarrollo en los que, sin necesidad de poseer un profundo conocimiento de sintaxis complejas, se pueden realizar aplicaciones completas. Ejemplos de estos entornos de programación son Delphi, de Borland, y Visual Basic, de Microsoft. En ocasiones, estas herramientas también se conocen como generadores automáticos de código fuente.

LA PROGRAMACIÓN ESTRUCTURADA

Una de las cuestiones más importantes que se debe tener en cuenta al desarrollar un programa en un lenguaje de alto nivel es el control de su ejecución. Son raros los programas que constan de un conjunto de instrucciones, que se ejecutan sucesivamente

una tras otra, y es frecuente tener que ejecutar varias veces algunas secciones del programa o bifurcar la ejecución en función del resultado de distintas condiciones.

Los lenguajes más antiguos (como Fortran) se apoyaban casi exclusivamente en una sola instrucción para definir el control de los programas: la instrucción GOTO (del inglés go to, que significa "ir a").

Tanto en Fortran como en BASIC, la instrucción GOTO y sus variantes condicionales van acompañadas de una o más etiquetas, que indican el número de la instrucción o línea de programa a la que debe dirigirse la transferencia de control. Pero las etiquetas, muchas veces numéricas, y las transferencias arbitrarias, hacen que los programas resulten muy poco legibles y difíciles de seguir. A finales de los años sesenta surgió una nueva forma de programar que reduce a la mínima expresión el uso de la instrucción GOTO y la sustituye por otras más comprensibles.



FIG. 16 El popular lenguaje C++ no es sino una extensión del lenguaje C original creado por Dennis Ritchie y Brian Kernighan.

Todo ello se basaba en un famoso teorema que afirma que cualquier programa puede escribirse utilizando únicamente las tres instrucciones de control siguientes:

- Un bloque secuencial de instrucciones. Es decir, una serie de instrucciones que se ejecutan sucesivamente.
- La instrucción condicional alternativa, de la forma "IF condición THEN instrucción1 ELSE instrucción2". Si la condición se cumple se ejecutará "instrucción1". En caso contrario se ejecutará "instrucción2". En forma abreviada, esta instrucción suele llamarse IF-THEN-ELSE.



FIG. 17 Java es un lenguaje orientado a objetos que no depende de la plataforma de ejecución.

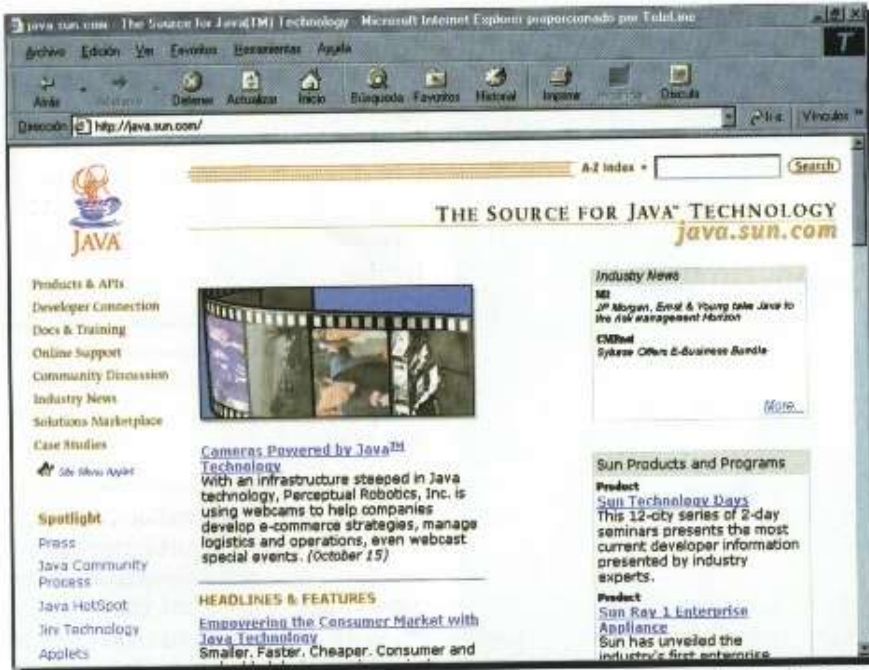


FIG. 18 Sun creó Java allá por el año 1991, y desde entonces este lenguaje ha cosechado éxito tras éxito.

■ El bucle condicional "WHILE condición DO instrucción", que ejecuta la instrucción repetidamente mientras la condición se cumpla. En su lugar, puede utilizarse también la forma "UNTIL condición DO instrucción", que ejecuta la instrucción hasta que la condición se cumpla.

Los programas que utilizan únicamente las tres instrucciones anteriores de control básicas o sus variantes (como los bucles FOR o la instrucción condicional CASE), pero que no usan la instrucción GOTO, se llaman "estructurados", y Pascal es el mejor ejemplo de lenguaje estructurado. La "programación estructurada" (llamada también

"programación sin GOTO") se convirtió durante la década de los años setenta en la forma de programar más extendida.

LOS LENGUAJES ORIENTADOS A OBJETOS

Más adelante, y como evolución de los lenguajes estructurados, aparecie-

ron los lenguajes orientados a objetos que, como en el caso de C++ o Java, no son otra cosa que una extensión del lenguaje C primitivo propuesto por Brian Kernighan y Dennis Ritchie.

La programación orientada a objetos u OOP (*Object Oriented Programming*) es una nueva forma de programar, que considera el programa como un conjunto de objetos que interactúan entre sí y que proliferó a partir de los años ochenta.

Como hemos visto anteriormente, la programación clásica basada en procedimientos planteaba ciertos problemas derivados del uso de cláusulas GOTO, problemas que se resolvieron con lo que se denominó programación estructurada, aunque, con el aumento de la complejidad de los programas dicho avance no fue suficiente para la mayoría de los programadores.

Hay que tener en cuenta que, a medida que se desarrollaban aplicaciones y sistemas informáticos más complejos y se utilizaba un mayor número de líneas de código, los problemas derivados del control del flujo, de la reutilización del código y de la legibilidad y mantenimiento de las aplicaciones, aumentaban también de forma considerable, haciendo que el proceso de programación resultara lento y arduo.

```
function makearray(n) {
    this.length = n;
    for(var i = 1; i <= n; i++)
        this[i] = 0;
    return this;
}

hexa = new makearray(10);
for(var i = 0; i < 10; i++)
    hexa[i] = i;
hexa[10] = "a"; hexa[11] = "b"; hexa[12] = "c";
hexa[13] = "d"; hexa[14] = "e"; hexa[15] = "f";

function hex(i) {
    if (i < 0)
        return "00";
    else if (i > 255)
        return "FF";
    else
```

FIG. 19 Java es descendiente directo de los lenguajes C y C++, de ahí su similitud.

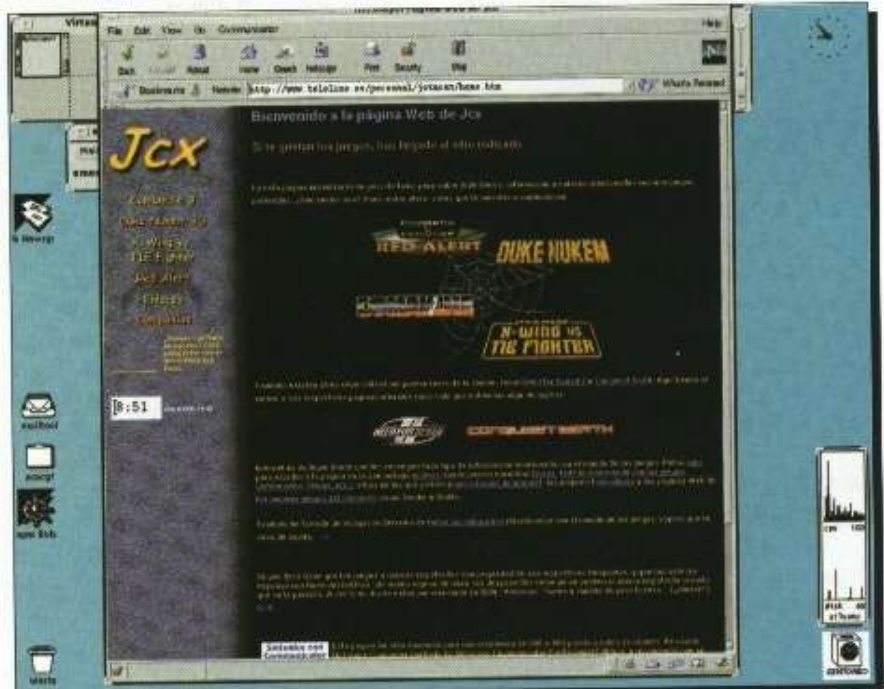


FIG. 20 Aquí podemos ver cómo una página Web con código Java se abre sin ningún problema en un sistema Solaris.



FIG. 21 El advenimiento de la WWW ha popularizado Java hasta límites insospechados.

La programación orientada a objetos (OOP) es una nueva forma de programar, que está especialmente pensada para salir al paso de los problemas indicados anteriormente.

Para ello se articula basándose en una serie de conceptos más o menos abstractos y que se manifiestan en una filosofía y una metodología de programación completamente diferentes a la que se utilizaba hasta el momento.

Estos conceptos de la programación orientada a objetos son los siguientes:

- **Objetos:** entidades complejas provistas de datos (propiedades, atributos) y comportamiento (funcionalidad, programas, métodos). Corresponden a los objetos del mundo real.
- **Clases:** conjuntos de objetos que comparten propiedades y comportamiento.
- **Herencia:** las clases no están aisladas, sino que se relacionan entre sí, formando una jerarquía de clasificación. Los objetos heredan las propiedades y el comportamiento de todas las clases a las que pertenecen.
- **Encapsulamiento:** cada objeto está aislado del exterior, es un módulo natural, y la aplicación entera se reduce a una recopilación o rompecabezas de objetos. El aislamiento protege los da-

tos asociados a un objeto contra su modificación por fragmentos de código no autorizados a acceder a ellos, eliminando efectos secundarios e interacciones.

- **Polimorfismo:** programas diferentes, asociados a objetos distintos, pueden compartir el mismo nombre, aunque el significado del programa varíe según el objeto al que se aplica.

Además, la programación orientada a objetos introduce nuevos conceptos, que a veces no son más que nuevos nombres aplicados a conceptos antiguos, ya conocidos. Entre ellos destacan:

- **Método:** es un programa asociado a un objeto (o a una clase de objetos), cuya ejecución se desencadena mediante un "mensaje".
- **Mensaje:** comunicación dirigida a un objeto, que le ordena que ejecute uno de sus métodos con ciertos parámetros que se le pasan al mismo tiempo.
- **Propiedad, atributo o variable:** nombre que reciben los datos asociados a objetos o a clases de objetos.

Para que un sistema de programación pueda considerarse orientado a objetos debe poseer objetos, clases de objetos que formen una estructura jerárquica (clases que sean subclases de otras), herencia de métodos y propiedades entre clases, subclases y los objetos que pertenecen a ellas.

En cualquier caso, estos conceptos básicos y muchos otros se irán desarrollando con detalle a lo largo de este curso, no sólo en la sección de "Conceptos de programación", sino también de una manera más práctica en la sección dedicada a cada uno de los lenguajes contenidos en los CD-ROM que constituyen el software operativo que se incluyen en esta obra.

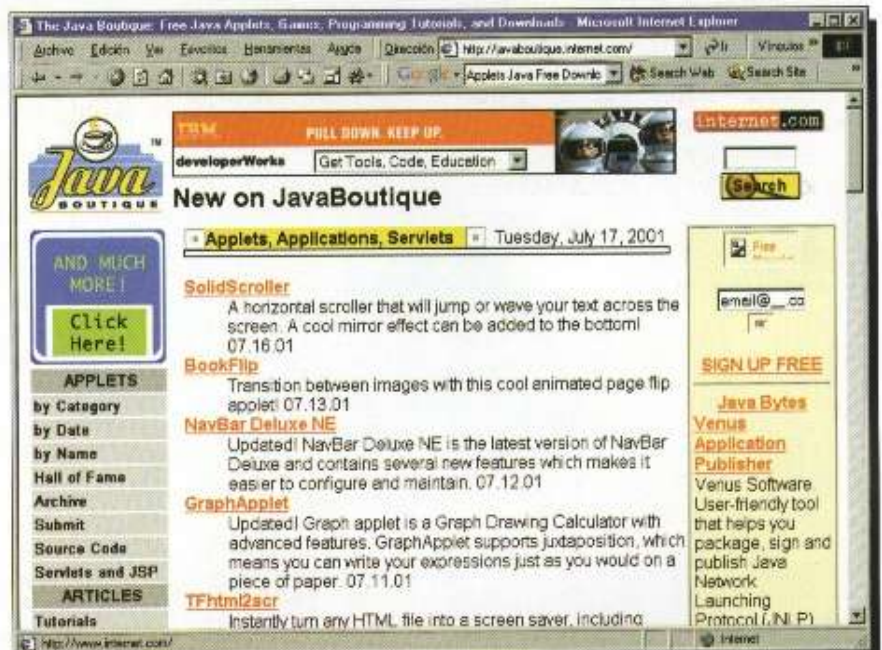


FIG. 22 Los applets de Java incluidos en una página Web se descargan desde Internet y se interpretan en la máquina cliente.

Plan de la obra

El Curso IBM Programar es fácil ha sido dirigido, coordinado y escrito por destacados especialistas en Informática, profesores de la Universidad Autónoma de Madrid, técnicos de IBM, especialistas en formación con técnicas multimedia del Instituto de Ingeniería del Conocimiento y editores de Multimedia Ediciones S.A.

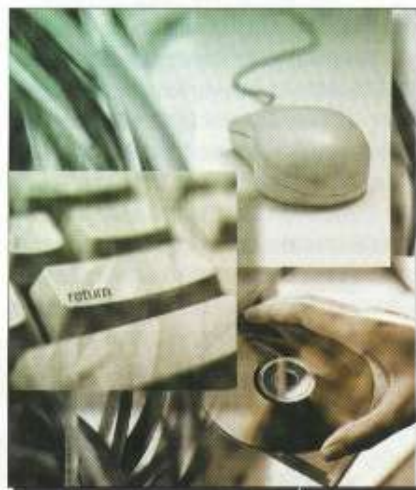
¿POR QUÉ UN CURSO DE PROGRAMACIÓN?

Cada día es más frecuente encontrar personas que dominan tanto el uso de los programas más habituales para PC (procesadores de textos, hojas de

cálculo, bases de datos y creación de gráficos), como la navegación por Internet o, incluso, la automatización (mediante el uso de macros) de algunas de las tareas más rutinarias que realiza con el procesador de textos o la

hoja de cálculo.

En esta situación se plantean preguntas del tipo: ¿estoy realmente muy lejos de ser capaz de desarrollar aplicaciones en mi PC que resuelvan mis necesidades diarias tanto en casa como en el trabajo? ¿estoy preparado para crear juegos, realizar una completa aplicación multimedia que integre imágenes, sonidos, textos y animaciones, o desarrollar mis propias aplicaciones para Internet?, ¿qué co-



nocimientos necesitaría para ello?, ¿es realmente complejo adquirir este nivel de conocimientos? ¿es la programación un feudo reservado de manera exclusiva a los profesionales? La respuesta a estas preguntas es que, en la actualidad, cualquier persona con unos conocimientos básicos del PC, puede satisfacer estas necesidades gracias a la programación.

Hoy en día ha desaparecido casi por completo la complejidad que la programación tenía en sus inicios gracias a los nuevos productos de programación visual. Realmente ahora le será muy fácil adquirir estos conocimientos que le permitirán ampliar sus expectativas en el entorno del PC. Este es el principal objetivo del Curso IBM Programar es fácil.



ORIENTACIÓN Y ESTRUCTURA DEL CURSO

Para el desarrollo del Curso IBM Programar es fácil se han seleccionado, con vistas a facilitar el aprendizaje tanto de los lenguajes de programación como de sus técnicas, los productos de programación visual de IBM y Borland.

Con la ayuda de este software y prácticamente sólo con el uso del ratón (habrá que escribir en realidad sólo una mínima parte del código) desarrollaremos diversos programas de aplicación similares a los que desarrollan los equipos profesionales.

Cada unidad didáctica, consta de tres secciones fijas. La primera trata de los conceptos teóricos y los distintos lenguajes de programación.

La segunda se centra en la programación más "doméstica" o de "emergencia" (p. ej. automatización de procesos sin necesidad de conocer lenguajes de programación).

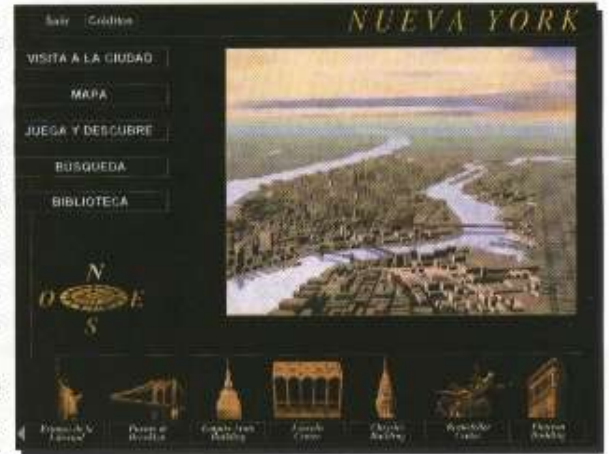
Finalmente, la tercera está

dedicada a la programación mediante el uso de lenguajes de programación.

Cada unidad incluye ejercicios prácticos de autoevaluación, con sus correspondientes respuestas, que le permitirán conocer en todo momento el nivel de conocimientos adquiridos.

Los programas que se incluyen en el curso, son todos de programación visual a excepción, obviamente, de algunas de las utilidades y herramientas tratadas en la sección de

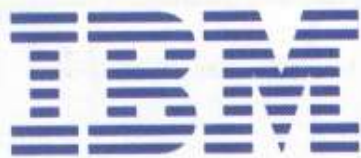
programación de "emergencia" y Borland Turbo Assembler. Los CD-ROM que componen la colección contienen además del software operativo, los programas fuente, la entrega correspondiente del Curso de Programación en formato PDF de alta resolución tratada en la tercera sección de la unidad didáctica, y los materiales de todas las aplicaciones que se desarrollan en el curso, alguna tan espectacular como la completa aplicación multimedia "Descubre Nueva York", para que pueda reutilizarlos en sus desarrollos.



LÍNEA DIRECTA

Además dispondrá de una línea directa de consulta que, a través del teléfono, fax o correo electrónico, le ayudará a solucionar las dudas que se le puedan plantear acerca de la instalación o el funcionamiento general de los programas.





Compañía líder del sector informático mundial. Está presente en más de 140 países con la gama de soluciones de informática y comunicaciones más completa del mercado.

Desde su fundación, en 1914, IBM está a la vanguardia tecnológica de esta industria lo que ha hecho que su esfuerzo de investigación y desarrollo esté en el origen de buena parte de las más importantes innovaciones registradas a lo largo de la historia de la informática.

Un esfuerzo centrado en la creación de soluciones tecnológicas que aporten el máximo valor y eficacia a las necesidades de los clientes. Tecnología para un mundo abierto, diverso y, cada vez más, interconectado.

Para más información acerca de IBM, visite <http://www.ibm.com>.



Borland International es el principal proveedor de productos de software de alta calidad para programadores y desarrolladores profesionales.

Borland se distingue por sus herramientas visuales para el desarrollo rápido de aplicaciones y su tecnología de conectividad entre ordenadores, tanto para sistemas de sobremesa, como cliente/servidor, Intranet/Internet y corporativos. Los productos de Borland cuentan con el apoyo de amplios programas para desarrolladores de empresas e independientes, distribuidores de valor añadido e integradores de sistemas. Fundada en 1983, Borland tiene la sede central en Scotts Valley, California.

Para más información sobre Borland, visite <http://www.borland.es>.



Siglas del Instituto de Ingeniería del Conocimiento. El IIC, creado en 1989, con sede en la Universidad Autónoma de Madrid, es uno de los más importantes centros españoles de I+D en el área de la Inteligencia Artificial y la Tecnología Multimedia.

El Instituto de Ingeniería del Conocimiento es pionero en el desarrollo de sistemas multimedia, fundamentalmente orientados hacia actividades de formación en los entornos IBM PC y Windows. En la actualidad, el IIC está a la cabeza de los desarrollos en esta área de actividad gracias a la incorporación de las más modernas técnicas, como la tecnología orientada a objetos o la inteligencia artificial.

Para más información acerca del IIC, visite <http://www.iic.uam.es>.



PROGRAMAR ES FÁCIL

El Curso IBM Programar es fácil, además de permitirle realizar sus propios programas para Windows, pondrá a su disposición el software de desarrollo más utilizado, así como una gran cantidad de ejemplos prácticos, incorporados en los CD-ROM, que podrá adaptar a sus propias aplicaciones.



47 CD-ROM



9 PROGRAMAS OPERATIVOS COMPLETOS (47 CD-ROM), ENTRE LOS MÁS PRESTIGIOSOS Y USADOS DEL MUNDO

IBM WEBSHERE STUDIO. Una potente herramienta de desarrollo para la creación de páginas Web en Internet e intranets para Windows.

BORLAND DELPHI. Un sencillo entorno con arquitectura orientada a objetos para realizar, con técnicas de programación visual, aplicaciones de 32 bits.

BORLAND C++BUILDER. La forma más potente de construir aplicaciones Windows combinando lo mejor de la programación visual con el código propio en lenguaje C y su extensión orientada a objetos C++.

IIC MULTIUS. Herramienta para el desarrollo de interfaces de usuario y aplicaciones multimedia, a partir de su descripción en un lenguaje propio que permite incluir sentencias en lenguaje C.

BORLAND TURBO ASSEMBLER. Compilador en lenguaje máquina para su utilización en ordenadores personales con entorno Windows.

BORLAND TURBO DEBUGGER. Herramienta de gran eficacia para corregir errores en los programas para Windows que vaya creando.

IBM DB2. Completo gestor de base de datos, diseñado para llevar a cabo, de forma fácil y con una potencia excepcional, la administración, el análisis y la producción de informes con sus datos.

IBM VISUALAGE JAVA. Completo entorno visual integrado que pone a su disposición todo el soporte necesario para el desarrollo de applets y programas en lenguaje Java.

BORLAND JBUILDER. La herramienta más rápida para desarrollo visual de aplicaciones 100% Java. Totalmente compatible con los estándares de la industria.

UN GLOSARIO INFORMATIZADO que contiene todos los términos informáticos utilizados a lo largo del Curso IBM Programar es fácil.

PROGRAMAS FUENTE Y MATERIALES de todas las aplicaciones que se desarrollan en el curso, para que pueda reutilizarlos en las que usted mismo cree.

PROGRAMAS Y UTILIDADES para la realización de la programación de "emergencia".

CURSO DE PROGRAMACIÓN (formato PDF), para consulta interactiva de los contenidos de la tercera sección de programación del curso.

47 UNIDADES



3 SECCIONES DONDE ENCONTRARÁ TODO LO QUE HAY QUE SABER SOBRE PROGRAMACIÓN

EN LA PRIMERA SECCIÓN, se tratan en detalle conceptos tan importantes como los algoritmos y estructuras de datos, punteros, stacks, listas, árboles, la programación estructurada, la multitarea, la programación orientada a objetos, así como una completa introducción a los distintos lenguajes de programación (HTML, Pascal, C, C++, Assembler y Java).

EN LA SEGUNDA SECCIÓN veremos como, conociendo una serie de utilidades y herramientas de uso muy sencillo, se pueden resolver multitud de problemas cotidianos o automatizar un buen número de procesos relativamente sencillos y repetitivos, sin necesidad de recurrir a una programación más seria y compleja. En esa sección nos ocuparemos de temas tan útiles como el desarrollo de macros del sistema, la programación de archivos batch para MS-DOS, la automatización de

procesos gráficos, el desarrollo de macros para Word, Excel y Access, la creación de archivos de ayuda de Windows, la gestión de la descarga de archivos desde Internet, la realización de programas de instalación y desinstalación, o los lenguajes de script para Internet hoy en día tan en boga.

LA TERCERA SECCIÓN estará dedicada a la programación mediante el uso de lenguajes de programación. Esta sección le permitirá, gracias a los programas operativos que se entregan en los CD-ROM, adentrarse de forma progresiva en el campo de la programación con los lenguajes más conocidos: HTML (con IBM WebSphere Studio), Pascal (con Borland Delphi), C y C++ —programación orientada a objetos— (con Borland C++Builder), ensamblador (con Borland Turbo Assembler) y Java (con IBM VisualAge y Borland JBuilder). También aprenderá a programar bases de datos (con IBM DB2) y aplicaciones multimedia (con IIC Multius). Utilizando a fondo estos programas desarrollará vistosas aplicaciones como un sitio Web sobre astronomía o la aplicación multimedia "Descubre Nueva York". Éstas y otras muchas de las aplicaciones que se desarrollarán durante el curso le permitirán crear con posterioridad cualquier programa de utilidad que pueda necesitar.

CURSO IBM PROGRAMAR ES FÁCIL

MULTIMEDIA EDICIONES S.A., empresa constituida por IBM y Planeta DeAgostini, líder en la realización de cursos y coleccionables que incluyen programas informáticos, ha creado, en colaboración con IBM, Borland y el Instituto de Ingeniería del Conocimiento, líderes en el sector del software, el



Con el que podrá dominar los más conocidos y actuales entornos de programación. Será capaz de desarrollar sus propias aplicaciones, y dejará de ser sólo un usuario del PC y conseguirá sacar el máximo rendimiento de su ordenador.

REQUERIMIENTOS HARDWARE

Microprocesador **INTEL PENTIUM** o compatibles a **200 MHz** o superior
Memoria mínima **64 MB**
Unidad lectora de **CD-ROM**
DISCO DURO
Tarjeta gráfica **SVGA** o superior
RATÓN compatible Microsoft

REQUERIMIENTOS SOFTWARE

Sistema operativo **MICROSOFT WINDOWS**
Los programas operativos, que tienen todas sus funcionalidades, se distribuyen de manera que usted pueda utilizarlos desde el primer momento.

Todos los CD-ROM del **CURSO IBM PROGRAMAR ES FÁCIL** disponen de un programa de instalación y desinstalación de uso muy sencillo. Los programas de instalación y desinstalación pueden ejecutarse cuantas veces se desee, por lo que es recomendable utilizar la opción de desinstalación para los programas que no se utilicen con frecuencia y, de esta forma, liberar espacio en el disco duro.

COMPOSICIÓN DEL CURSO

Cada semana recibirá una unidad didáctica acompañada por su correspondiente CD-ROM

47 UNIDADES DIDÁCTICAS de periodicidad semanal
Encuadernables en 4 volúmenes.
948 páginas.
470 ejercicios de autoevaluación.
Más de 2.000 gráficos, capturas de pantallas, esquemas, etc.

47 CD-ROM que contienen:
9 programas operativos completos (46 CD-ROM).
Un completo glosario de términos informáticos (1 CD-ROM).
Programas fuente y materiales de todas las aplicaciones que se desarrollan a lo largo del curso.
Una completa colección de utilidades.
Curso de programación (en formato PDF).

Todos los programas operativos incluyen una completa función de ayuda en línea/on-line.
Opción a obtener un Diploma de estudios, personalizado y garantizado por IBM España.
Línea directa de consulta por teléfono, fax y correo electrónico (e-mail).

Estructura de las unidades didácticas



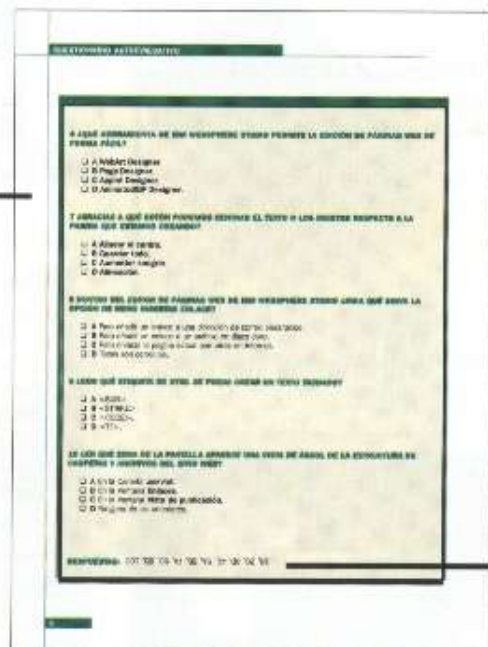
Los temas básicos de la programación actual explicados con un lenguaje ameno y riguroso y tratados mediante multitud de ejemplos.

Reproducción exacta de las pantallas de los programas que permiten seguir, con total comodidad y paso a paso, el funcionamiento de las aplicaciones de los CD-ROM que acompañan a las unidades didácticas.



Recuadros que resumen, completan o amplían los conceptos o indicaciones del texto. Estos recuadros facilitan la lectura de los principales puntos tratados en la unidad y redondean la eficacia de la obra.

El cuestionario autoevaluativo que acompaña a cada unidad del curso es un instrumento de gran utilidad para valorar el grado de asimilación de los temas tratados.



Las respuestas a las preguntas claras y concisas del cuestionario autoevaluativo que se encuentran al pie de la página se han desarrollado en las explicaciones, recuadros e ilustraciones de la propia unidad.

Cada unidad se ha diseñado contando con una dedicación semanal de 10 horas, con las prácticas y los ejercicios incluidos

IBM WebSphere Studio

Programación visual para redes Internet/Intranet

6 CD-ROM con las unidades 1 a 6

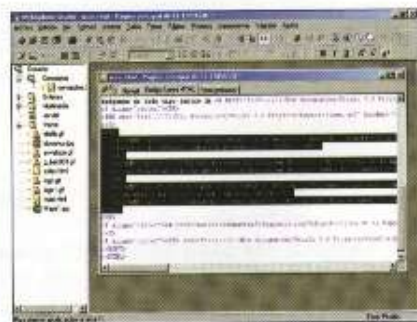
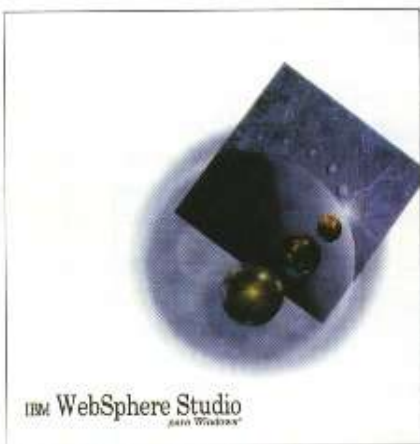
Un producto



Crear un sitio Web no es sencillo, sobre todo si el objetivo no es sólo diseñar un espacio de ocio personal, sino desarrollar un entorno Web que permita que una empresa entre por la puerta grande de Internet. IBM WebSphere Studio es la herramienta ideal tanto para crear sitios Web personales como para los más profesionales.

INCLUYE:

- Visualización gráfica de los enlaces entre los archivos de un proyecto.
- Actualización totalmente automática de los enlaces al modificar o mover los archivos.
- Compatibilidad con sus herramientas favoritas para editar los archivos de un proyecto.
- Creación y edición visual de archivos HTML y JSP.
- Creación de páginas Web dinámicas utilizando bases de datos y Java Beans mediante los asistentes que incorpora.
- Posibilidad de montar y publicar sus Web en varios servidores.
- Compatibilidad con las bases de datos más conocidas.
- Depuración de código JSP y Java en tiempo real.
- Opciones para montar applets de Java, crear animaciones GIF, imágenes de cabecera, botones y gráficos para sus páginas Web con las herramientas complementarias que incorpora.
- Desarrollo de proyectos de gran complejidad (hasta 250 archivos por proyecto).



Borland Delphi

Programación visual en Pascal

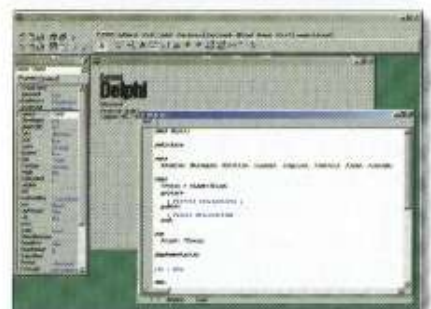
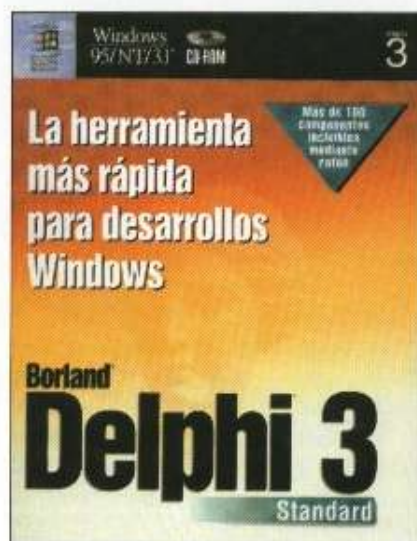
6 CD-ROM con las unidades 7 a 12

Un producto
Borland

Borland Delphi es un entorno de programación que goza de una amplia aceptación en el mercado profesional ya que permite una forma muy visual de trabajar en Programación Orientada a Objetos para desarrollar aplicaciones de 32 bits.

INCLUYE:

- Alto rendimiento en el desarrollo de aplicaciones en 32 bits gracias a su compilador propio.
- Inclusión de código propio en lenguaje Pascal.
- Soporte completo de Windows.
- Almacenamiento de objetos reutilizables en otras aplicaciones.
- Biblioteca de componentes visuales con más de 100 componentes reutilizables.
- Entorno integrado de desarrollo, incluyendo depuración y seguimiento, así como mensajes de error de fácil comprensión.
- Asistentes para agilizar la codificación de DLLs y EXECs.
- Controladores y servidores de automatización OLE.
- Almacén de objetos y herencia de fichas visuales que permiten reutilizar el código.
- Compatibilidad y conectividad para múltiples bases de datos.



Borland C++ Builder

Programación visual en C y C++

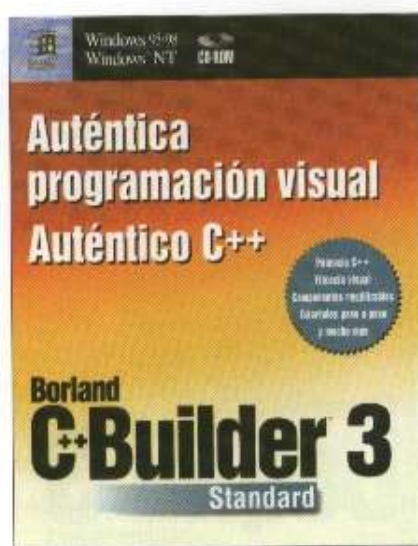
6 CD-ROM con las unidades 13 a 18

Un producto
Borland

C++ es un lenguaje de programación que ha sabido adaptarse a los nuevos tiempos. Es con diferencia el más importante de todos los que se emplean en el desarrollo profesional de aplicaciones y, gracias a productos como Borland C++ Builder, es capaz de ofrecer un entorno sin par de programación visual.

INCLUYE:

- Toda la potencia de C++, con la tecnología de compilación de C++ de Borland, y con el aumento de velocidad de su nuevo "linker" incremental.
- Entorno integrado de desarrollo muy flexible.
- Depurador y editor integrados.
- Biblioteca de componentes visuales, con más de 100 componentes para aumentar la productividad en el desarrollo de programas.
- Compilación de programas ANSI C y C++.
- Compatibilidad total con los estándares de Microsoft y de otros fabricantes (ANSI C++, Microsoft Windows, Win32 API, OLE, Microsoft Access, Multibyte character, etc.).
- Ayuda en línea sensible al contexto mediante hipertextos.
- Desarrollo cómodo y rápido de potentes aplicaciones de bases de datos con herramientas visuales y el sólo uso del ratón.
- Reutilización de los formularios, componentes, módulos de datos y código de Borland Delphi.



IIC Multius

Programación de aplicaciones multimedia

5 CD-ROM con las unidades 19 a 23

Un producto

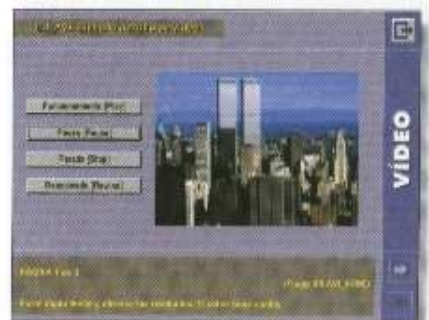


A diferencia de otros programas similares, IIC Multius destaca por la sencillez con la que se pueden integrar, modificar y actualizar los objetos multimedia. Además, dispone de un potente lenguaje propio (MDL) que pone a disposición del usuario una forma muy sencilla de construir páginas y de definir componentes de Windows.



INCLUYE:

- Lenguaje MDL de fácil uso, que mantiene de forma separada la lógica de la aplicación con las operaciones y los objetos que incluye (texto, imágenes, sonido, animaciones, etc.), lo que simplifica el mantenimiento posterior de aplicaciones y su adaptación a otros idiomas.
- Ayuda interactiva para el diseño de las pantallas de la aplicación.
- Tecnología de programación orientada a objetos mediante el concepto de "widgets", que corresponden a todo objeto que puede aparecer en una página o pantalla de la aplicación.
- Una completa librería de "widgets" ya programados que permiten el desarrollo de aplicaciones multimedia con un mínimo de conocimientos de programación.
- Además de los "widgets" asociados a los controles estándar de Windows, se incluyen otros más potentes, como son los de presentación de bitmaps, sonido, animaciones, hipertexto, vídeo y botones con bitmaps.
- Código final de la aplicación en C, compilable con Borland C++ Builder, lo que garantiza su transportabilidad y eficiencia.



Borland Turbo Assembler

Programación en ensamblador

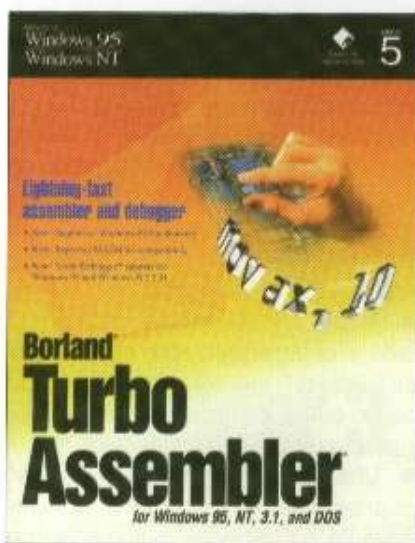
3 CD-ROM con las unidades 24 a 26

Un producto
Borland

Aunque muchos puedan pensar que, hoy en día, la programación se realiza usando exclusivamente herramientas visuales, esto no es totalmente cierto. Todavía se necesitan programas como Borland Turbo Assembler, un compilador de ensamblador capaz de sacar el máximo provecho de la máquina.

INCLUYE:

- Desarrollo de aplicaciones para 16 y 32 bits muy eficientes en cuanto al tiempo de respuesta.
- Desarrollo de módulos integrables con aplicaciones escritas en otros lenguajes de más alto nivel (C, C++, Pascal, etc.).
- Soporte total de los distintos microprocesadores de su ordenador, desde el 8088 al Pentium.
- Fácil reutilización de código mediante el uso de sus extensiones orientadas a objetos.
- Desarrollo de código robusto con su innovador modo de ensamblar (IDEAL).
- Ensamblador para más de 48.000 líneas de código por minuto.
- Depuración de errores con Borland Turbo Debugger (incluido).



IBM DB2

Gestión de bases de datos

7 CD-ROM con las unidades 27 a 33

Un producto



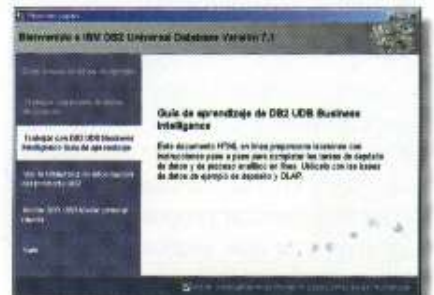
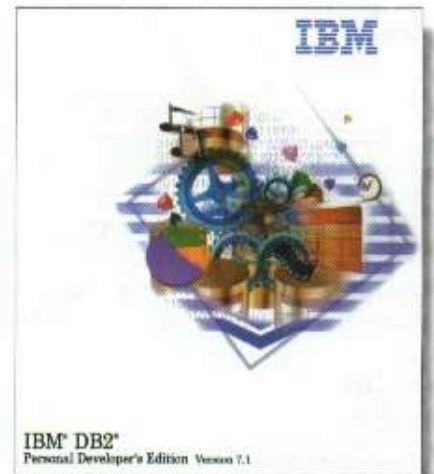
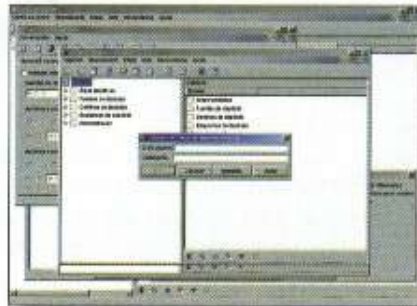
Hoy en día la mayor parte de los desarrollos que se realizan en la empresa incluyen la programación de bases de datos. Esta es la razón por la que se incluye en el curso el programa IBM DB2, un completo gestor de base de datos de gran implantación en el ámbito empresarial.

INCLUYE:

- Potencia relacional. Con IBM DB2 podrá administrar fácilmente información y vincular distintas bases de datos. IBM DB2 incluye una completa colección de operadores y funciones para aumentar la potencia de sus cálculos.
- Completa gestión de datos: tablas, índices, consultas, etc.
- Utilización por usuarios individuales y también en entorno multiusuario.
- Lenguaje estructurado SQL especialmente diseñado para las tareas de gestión de base de datos.
- Mecanismos de administración, seguridad e integridad de las bases de datos.
- Procedimientos almacenados, "triggers" y alertas.
- Conectividad garantizada con otras aplicaciones (estándar de base de datos ODBC).
- Centro de comandos desde el que puede ejecutar interactivamente sentencias SQL sobre sus bases de datos.
- Mecanismos para acceder a múltiples bases de datos al mismo tiempo.
- Precompiladores que permiten el desarrollo de aplicaciones en lengua-

jes de alto nivel (C/C++, Java, COBOL y Fortran) incluyendo sentencias SQL.

- Transacciones y acceso concurrente a datos.
- Programación dinámica mediante la construcción de las sentencias SQL en tiempo de ejecución.



IBM VisualAge for Java

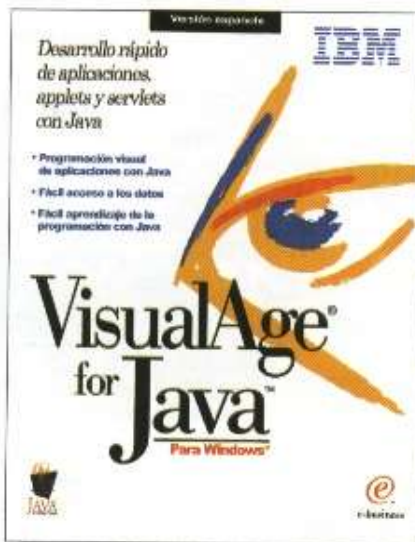
Programación visual en Java

7 CD-ROM con las unidades 34 a 40

Un producto



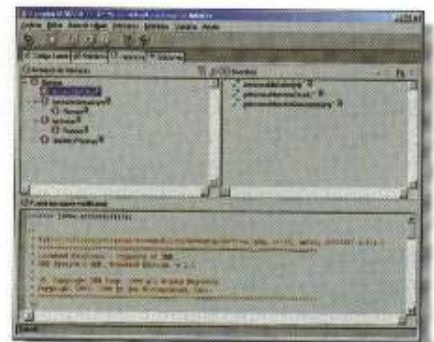
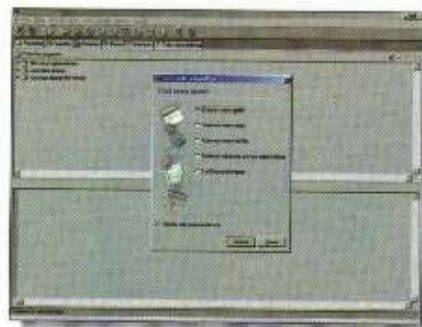
IBM VisualAge for Java es un entorno visual integrado que brinda todo el soporte necesario para el desarrollo de applets y programas en lenguaje Java. Aunque este producto es especial por muchas de sus características, una de las más sobresalientes es sin duda su orientación a la edición visual.



- Creación de características para JavaBeans.
- Importación y exportación de código Java.
- Depurador de código Java totalmente visual.
- Herramientas para crear, modificar y utilizar JavaBeans.
- Sistema de gestión de código para el mantenimiento de múltiples versiones de un programa.
- Entorno de prueba para la depuración de componentes Enterprise JavaBeans y JavaServer Pages.
- Examinadores personalizables y asistente de código ampliado en el entorno integrado de desarrollo (IDE), incluyendo soporte de macros.

INCLUYE:

- Editor de composición visual para diseñar la interfaz de usuario del programa, especificar su comportamiento y definir su relación con el resto del programa.
- Generación totalmente visual de código Java. En muchos casos es posible diseñar y ejecutar programas completos sin tener que escribir una sola línea de código.
- Asistentes (SmartGuides) para la creación de applets, proyectos, paquetes, clases, interfaces y métodos.



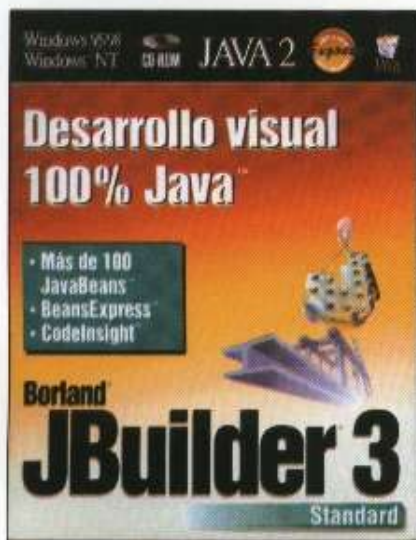
Borland JBuilder

Programación visual en Java

6 CD-ROM con las unidades 41 a 46

Un producto
Borland

El término "Web" es, hoy por hoy, sinónimo de "Java". Sin embargo, aunque Java haya alcanzado una gran popularidad gracias al auge de la World Wide Web (WWW), no hay que olvidar que se trata de un potente lenguaje de programación multiplataforma que se extiende con rapidez por el mercado gracias a sus similitudes con C.



INCLUYE:

- Más de 50 módulos, componentes JavaBean, ya programados reutilizables en nuevas aplicaciones, incluyendo botones, casillas de verificación, listas, barras de menú, barra de desplazamiento, menús desplegables, campos de texto, diseños, etc.
- Productividad inmediata mediante el uso de asistentes para simplificar el desarrollo de aplicaciones.
- Visualizador de aplicaciones, combinando las necesidades de un gestor de proyectos, de un visualizador de clases y ficheros y un editor de código.
- Entorno integrado de desarrollo rápido de aplicaciones (RAD) que incorpora, visualizador de aplicaciones, paleta de componentes visuales, diseñadores visuales, visualizador HTML, depurador gráfico y compilador.
- Compatibilidad total con los estándares de la industria: Java 2, JDK 1.1, JavaBeans, JFC, archivos JAR, Clases Internas, Internacionalización, Seguridad y JNI entre otros.
- Depurador gráfico para Java 2.
- CodeInsight para agilizar la codificación y reducir los errores de sintaxis.

