

Lenguajes de alto nivel

Bernardo Beltrán Bernáldez N° 3
Eusebio Peña Toril N° 14

ÍNDICE

INTRODUCCIÓN	3
BASIC	4
COBOL	5
PASCAL	5
FORTRAN	6
LISP	6
ADA	7
APL	7
PL/1	7
RPG	8
Lenguaje C	8
Lenguaje C++	9
C#	9
Visual BASIC	9
ALGOL	11
Ejemplo de programa en Algol	11
JAVA	12
Léxico	13
Postscript	13
Perl	14
FORTH	14
PROLOG	15
HASKELL	15
MODULA-2	16
PYTHON	16
PL/SQL	17
PHP	18

INTRODUCCIÓN

Los lenguajes de alto nivel suelen utilizar términos ingleses del tipo LIST, PRINT u OPEN como comandos que representan una secuencia de decenas o de centenas de instrucciones en lenguaje máquina. Los comandos se introducen desde el teclado, desde un programa residente en la memoria o desde un dispositivo de almacenamiento, y son interceptados por un programa que los traduce a instrucciones en lenguaje máquina.

Los programas traductores son de dos tipos: intérpretes y compiladores. Con un intérprete, los programas que repiten un ciclo para volver a ejecutar parte de sus instrucciones, reinterpretan la misma instrucción cada vez que aparece. Por consiguiente, los programas interpretados se ejecutan con mucha mayor lentitud que los programas en lenguaje máquina. Por el contrario, los compiladores traducen un programa íntegro a lenguaje máquina antes de su ejecución, por lo cual se ejecutan con tanta rapidez como si hubiesen sido escritos directamente en lenguaje máquina.

Se considera que fue la estadounidense Grace Hopper quien implementó el primer lenguaje de ordenador orientado al uso comercial. Después de programar un ordenador experimental en la Universidad de Harvard, trabajó en los modelos UNIVAC I y UNIVAC II, desarrollando un lenguaje de alto nivel para uso comercial llamado FLOW-MATIC. Para facilitar el uso del ordenador en las aplicaciones científicas, IBM desarrolló un lenguaje que simplificaría el trabajo que implicaba el tratamiento de fórmulas matemáticas complejas. Iniciado en 1954 y terminado en 1957, el FORTRAN (acrónimo de Formula Translator) fue el primer lenguaje exhaustivo de alto nivel de uso generalizado.

En 1957 una asociación estadounidense, la Asociación for Computing Machinery comenzó a desarrollar un lenguaje universal que corrigiera algunos de los defectos del FORTRAN. Un año más tarde fue lanzado el ALGOL (acrónimo de Algorithmic Language), otro lenguaje de orientación científica. De gran difusión en Europa durante las décadas de 1960 y 1970, desde entonces ha sido sustituido por nuevos lenguajes, mientras que el FORTRAN continúa siendo utilizado debido a las gigantescas inversiones que se hicieron en los programas existentes. El COBOL (acrónimo de Common Business Oriented Language) es un lenguaje de programación para uso comercial y empresarial especializado en la organización de datos y manipulación de archivos, y hoy día está muy difundido en el mundo empresarial.

El lenguaje BASIC (acrónimo de Código de Instrucciones Simbólicas de Uso General para Principiantes) fue desarrollado en el Dartmouth College a principios de la década de 1960 y está dirigido a los usuarios de ordenador no profesionales. Este lenguaje se universalizó gracias a la popularización de los microordenadores en las décadas de 1970 y 1980. Calificado de lento, ineficaz y poco estético por sus detractores, BASIC es sencillo de aprender y fácil de utilizar. Como muchos de los primeros microordenadores se vendieron con BASIC incorporado en el hardware (en la memoria ROM), se generalizó el uso de este lenguaje.

Aunque existen centenares de lenguajes informáticos y de variantes, hay algunos dignos de mención, como el PASCAL, diseñado en un principio como herramienta de enseñanza, hoy es uno de los lenguajes de microordenador más populares; el Logo fue desarrollado para que los niños pudieran acceder al mundo de la informática; el C, un lenguaje de Bell Laboratories diseñado en la década de 1970, se utiliza ampliamente en el desarrollo de programas de sistemas, al igual que su sucesor, el C++. El LISP y el PROLOG han alcanzado amplia difusión en el campo de la inteligencia artificial

Son aquellos que se encuentran más cercanos al lenguaje natural que al lenguaje

máquina.

Están dirigidos a solucionar problemas mediante el uso de EDD's.

Nota: EDD's son las abreviaturas de Estructuras Dinámicas de Datos, algo muy utilizado en todos los lenguajes de programación. Son estructuras que pueden cambiar de tamaño durante la ejecución del programa. Nos permiten crear estructuras de datos que se adapten a las necesidades reales de un programa.

Se tratan de lenguajes independientes de la arquitectura del ordenador. Por lo que, en principio, un programa escrito en un lenguaje de alto nivel, lo puedes migrar de una máquina a otra sin ningún tipo de problema.

Estos lenguajes permiten al programador olvidarse por completo del funcionamiento interno de la máquina/s para la que están diseñando el programa. Tan solo necesitan un traductor que entienda el código fuente como las características de la máquina.

Suelen usar tipos de datos para la programación y hay lenguajes de propósito general (cualquier tipo de aplicación) y de propósito específico (como FORTRAN para trabajos científicos).

BASIC (Beginners All-purpose Symbolic Instruction Code)

Fue el lenguaje de programación interactivo más popular en la década de los 70. Es un lenguaje de propósito general. Desarrollado por *John Kemeny* y *Thomas Kurtz* en "Dartmouth College" en 1963. Existen numerosas versiones, algunas son compiladoras y otras son intérpretes. BASIC es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo, estructuras sencillas y un buen conjunto de operadores. Es un lenguaje que no está especializado en ningún tipo de aplicación. Esto lo hace un lenguaje versátil y potente, con un campo de aplicación ilimitado y, sobre todo, se puede aprender rápidamente. En poco tiempo, un programador puede utilizar la totalidad del lenguaje.

La palabra BASIC proviene de la expresión inglesa *Beginner's All-purpose Symbolic Instruction Code*: código de instrucciones simbólicas de propósito general para principiantes.

El BASIC fue el primer lenguaje de programación desarrollado. Lo fue a mediados de la década de los sesenta por los profesores John G. Kemeny y Thomas E. Kurtz en el Dartmouth College, en California. Su código se basa en el vocabulario inglés y en las expresiones matemáticas. Consta de cientos de instrucciones para utilización de gráficos, procesado de textos, uso de variables, archivos, etc. EL BASIC utiliza un alfabeto formado por los caracteres alfabéticos: A-Z, cifras 0-9, caracteres especiales como operadores aritméticos: +, -, *, etc., y otros: (,), \$, etc.

El lenguaje BASIC es, originalmente, un lenguaje interpretado. Existen dos tipos de lenguaje: interpretados y compilados. Los interpretados son aquellos que necesitan del código fuente para funcionar (por ejemplo, GW-BASIC y QBasic). Los compilados, convierten el código fuente en un archivo objeto y éste en un archivo ejecutable. Este es el caso del lenguaje C y de las versiones más completas y recientes del BASIC.

Existen muchas versiones (implementaciones) del lenguaje BASIC. En el esquema siguiente se puede observar la evolución del desarrollo de los diferentes lenguajes de programación y, entre ellos, la de BASIC.

COBOL (Common Business Oriented Language)

Es un lenguaje compilador diseñado para aplicaciones de negocios. Desarrollado en 1959 por el gobierno federal de los Estados Unidos y fabricantes de computadoras bajo el liderazgo de *Grace Hopper*. Es el más utilizado por los "mainframe". COBOL está estructurado en cuatro divisiones; a saber:

División de identificación - identifica el programa.

División ambiental - identifica a las computadoras fuente y objeto.

División de datos - identifica las memorias "buffer", constantes y áreas de trabajo.

División de procedimiento - describe el procesamiento (la lógica del programa).

PASCAL

Este programa recibió su nombre en honor a Blaise Pascal. Fue desarrollado por el científico suizo Niklaus Wirth en 1970 y diseñado para enseñar técnicas de programación estructurada. Es fácil de aprender y de usar y no utiliza línea sino ";" (semicolon). Existen versiones de compilador, como de intérprete. Estas varían según la versión. A diferencia de lenguajes de programación descendientes de C, Pascal utiliza el símbolo:= para la asignación en vez de =. Si bien el segundo es más conciso, la práctica ha demostrado que muchos usuarios utilizan el símbolo de igualdad para comparar valores en lugar del comparador de C que es el símbolo ==. Esta sintaxis conduce a muchos errores o *bugs* difíciles de rastrear en código C. Dado que Pascal no permite dentro de expresiones y utiliza sintaxis distinta para asignaciones y comparaciones, no sufre estos errores.

Otra diferencia importante es que en Pascal, el tipo de una variable se fija en su definición; la asignación a variables de valores de tipo incompatible no están autorizadas (En C, en cambio, el compilador hace el mejor esfuerzo para dar una interpretación a casi todo tipo de asignaciones). Esto previene errores comunes donde variables son usadas incorrectamente porque el tipo es desconocido. Esto también evita la necesidad de notación húngara, esto es prefijos que se añaden a los nombres de las variables y que indican su tipo.

FORTRAN (Fórmula Translator)

Es uno de los primeros lenguajes de alto nivel desarrollado en 1954 por John Backus y un grupo de programadores de IBM. Es un lenguaje compilador que se diseñó para expresar con facilidad las fórmulas matemáticas, resolver problemas científicos y de ingeniería. FORTRAN

FORTRAN (formula translation) es un lenguaje desarrollado en 1957 por John Backus, de IBM, cuyas versiones más conocidas son la de FORTRAN IV y FORTRAN 77 (Fortran Estructurado). Recordemos que la computadora nació de la necesidad de resolver problemas matemáticos y científicos.

A mediados de la década de los 60's FORTRAN pasó a ser ampliamente conocido y usado en buen número de máquinas, dando como resultado una variedad de versiones de las cuales las más importantes fueron FORTRAN II y FORTRAN IV, ambos de IBM.

Las ventajas de este lenguaje de programación es que maneja variables de doble precisión, variables complejas y modularidad o subrutinas.

LISP

Después de Fortran es el segundo lenguaje de programación de alto nivel. LISP fue creado por John McCarthy y sus colaboradores en el Instituto de Tecnología de Massachusetts (M.I.T.). El elemento fundamental en LISP es la lista, en el sentido más amplio del término, pues tanto los datos como los programas son listas. De ahí viene su nombre, pues LISP es un acrónimo de "LIStProcessing".

Los lenguajes de este tipo se llaman "aplicativos" o "funcionales", porque se basan en la aplicación de funciones a sus datos.

En LISP se distinguen dos tipos fundamentales de elementos:

Átomos: son datos elementales y pueden pertenecer a varios tipos: números, caracteres, cadenas de caracteres y símbolos.

Listas: son secuencias de átomos o de listas encerradas entre paréntesis. Además, existe una lista especial, "nil", que es la lista nula, que no tiene ningún elemento.

En LISP, una función se expresa como una lista.

Algunas de las funciones predefinidas de LISP tienen símbolos familiares (+ para la suma, * para el producto), pero otras son más exóticas, especialmente dos que sirven precisamente para manipular listas, descomponiéndolas en sus componentes. Sus nombres ("car" y "cdr") son un poco extraños, reliquias de tiempos pasados y de la estructura de los ordenadores de segunda generación.

ADA

Es un lenguaje desarrollado como una norma del Departamento de Defensa de los Estados Unidos. Es un lenguaje basado en PASCAL, pero más amplio y específico. Fue diseñado tanto para aplicaciones comerciales como científicas. Es un lenguaje de multitareas que puede ser compilado por segmentos separados. Se llama ADA en honor de Augusta Ada Byron, condesa de Lovelace e hija del poeta inglés Lord Byron. Ada es un lenguaje de programación estructurado y fuertemente tipado de forma estática que fue diseñado por Jean Ichbiah de CII Honeywell Bull por encargo del Departamento de defensa de los Estados Unidos (DoD). Es un lenguaje multipropósito, orientado a objetos y concurrentes, pudiendo llegar desde la facilidad de Pascal hasta la flexibilidad de C++.

Fue diseñado con la seguridad en mente y con una filosofía orientada a la reducción de errores comunes y difíciles de descubrir. Para ello se basa en un tipado muy fuerte y en chequeos en tiempo de ejecución (desactivables en beneficio del rendimiento). La sincronización de tareas se realiza mediante la primitiva rendezvous.

Ada se usa principalmente en entornos en los que se necesita una gran seguridad y confiabilidad como la defensa, la aeronáutica (Boeing o Airbus), la gestión del tráfico aéreo (como Indra en España) y la industria aeroespacial entre otros.

APL (A Programming Language)

Este programa fue desarrollado por Kenneth Inverson a mediados de la década de 1960 para resolver problemas matemáticos. Este lenguaje se caracteriza por su brevedad y por su capacidad de generación de matrices y se utiliza en el desarrollo de modelos matemáticos.

PL/1 (Programming Language 1)

Este programa fue desarrollado por IBM. Es un lenguaje de propósito general que incluye características de COBOL y de FORTRAN. Su principal utilidad es en los "mainframes". En los primeros años de la década de los 60's, el programador podía elegir entre varios lenguajes.

PL/1 o Lenguaje de Programación #1 fue desarrollado por IBM a mediados de la década de los 60's, en un intento (y en eso se quedó) por crear un lenguaje que fuera adecuado para programar *cualquier* clase de aplicación.

Esto dio como resultado, por supuesto, un lenguaje que es enorme en todos los sentidos y que no es fácil de aprender en su totalidad.

RPG (Report Program Generator)

Fue desarrollado por IBM en 1964 y diseñado para generar informes comerciales o de negocios. En 1960 RPG es creado para la familia 1400, pero hasta 1964 no es lanzada la versión final para la IBM 360. Ha sido actualizado en diversas ocasiones, dando origen a las diferentes versiones del lenguaje. La última actualización que se ha realizado hasta la fecha es el RPG/IV en 1995, disponible con los ordenadores IBM de la familia AS/400.

LENGUAJE C

Fue desarrollado a principios de la década de los 70 en Bell Laboratories por *Brian Kernigham* y *Dennis Ritchie*. Ellos necesitaban desarrollar un lenguaje que se pudiera integrar con UNIX, permitiendo a los usuarios hacer modificaciones y mejoras fácilmente. Fue derivado de otro lenguaje llamado BCPL. Al igual que B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.

Se trata de un lenguaje débilmente tipado de medio nivel pero con muchas características de bajo nivel. Dispone de las estructuras típicas de los lenguajes de alto nivel pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.

La primera estandarización del lenguaje C fue en ANSI, con el estándar X3.159-1989. El lenguaje que define este estándar fue conocido vulgarmente como ANSI C. Posteriormente, en 1990, fue ratificado como estándar ISO (ISO/IEC 9899:1990). La adopción de este estándar es muy amplia por lo que, si los programas creados lo siguen, el código es portable entre plataformas y/o arquitecturas. En la práctica, los programadores suelen usar elementos no-portables dependientes del compilador o del sistema operativo

LENGUAJE C++

Se pronuncia "ce plus plus". Fue desarrollado por Bjarne Stroustrup en los Bell Laboratories a principios de la década de los '80. C++ introduce la programación orientada al objeto en C. Es un lenguaje extremadamente poderoso y eficiente. C++ es un súper conjunto de C, para aprender C++ significa aprender todo acerca de C, luego aprender programación orientada al objeto y el uso de éstas con C++.

C#

C# (pronunciado "si sharp" o **C sostenido**) es un lenguaje de programación orientado a objetos desarrollado por Microsoft y estandarizado, como parte de su plataforma .NET.

Su sintaxis básica deriva de C/C++ y utiliza el modelo de objetos de la plataforma .NET el cual es similar al de Java aunque incluye mejoras derivadas de otros lenguajes. C# fue diseñado para combinar el control a bajo nivel de lenguajes como C y la velocidad de programación de lenguajes como Visual Basic.

C# significa, "do sostenido" (C corresponde a do en la terminología musical anglo-sajona). El símbolo # viene de sobreponer "++" sobre "++" y eliminar las separaciones, indicando así su descendencia de C++.

C#, como parte de la plataforma .NET, está normalizado por ECMA desde diciembre de 2001 (ECMA-334 "Especificación del Lenguaje C#"). El 7 de noviembre de 2005 acabó la beta y salió la versión 2.0 del lenguaje que incluye mejoras tales como tipos genéricos, métodos anónimos, iteradores, tipos parciales y tipos anulables. Ya existe la versión 3.0 de C# en fase de beta destacando los tipos implícitos y el LINQ (Language Integrated Query).

Aunque C# forma parte de la plataforma .NET, ésta es una interfaz de programación de aplicaciones; mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Aunque aún no existen, es posible implementar compiladores que no generen programas para dicha plataforma, sino para una plataforma diferente como Win32 o UNIX

VISUAL BASIC

Este programa fue creado por Microsoft. Es un programa moderno que da apoyo a las características y métodos orientados al objeto. Es un lenguaje de fácil aprendizaje (pero algunos son más complicados debido a que la persona que lo usa tenga conocimiento de este programa amplio) pensado tanto para programadores principiantes como expertos, guiado por eventos, y centrado en un motor de formularios poderoso que facilita el rápido desarrollo de aplicaciones gráficas. Su principal innovación, que luego fue adoptada por otros lenguajes, fue el uso de un tipo de dll, llamado inicialmente vbx y

posteriormente ocx, que permiten contener toda la funcionalidad de un control y facilitar su rápida incorporación a los formularios.

Su sintaxis, derivada del antiguo BASIC, ha sido ampliada con el tiempo al agregarse las características típicas de los lenguajes estructurados modernos. Se ha agregado una implementación limitada de la Programación Orientada a Objetos (los propios formularios y controles son objetos), aunque si que admite el polimorfismo mediante el uso de los Interfaces no admite la herencia. No requiere de manejo de punteros y posee un manejo muy sencillo de cadenas de caracteres. Posee varias bibliotecas para manejo de bases de datos, pudiendo conectar con cualquier base de datos a través de ODBC (Informix, DBase, Access, MySQL, SQL Server, PostgreSQL ,etc.) a través de ADO.

Es utilizado principalmente para aplicaciones de gestión de empresas, debido a la rapidez con la que puede hacerse un programa que utilice una base de datos sencilla, además de la abundancia de programadores en este lenguaje.

El compilador de Microsoft genera ejecutables que requieren una DLL para que sus ejecutables funcionen, en algunos casos llamada MSVBVMxy.DLL (acrónimo de "MicroSoft Visual Basic Virtual Machine x.y", siendo x.y la versión) y en otros VBRUNXXX.DLL ("Visual Basic Runtime X.XX"), que provee todas las funciones implementadas en el lenguaje. Además existen un gran número de bibliotecas (DLL) que facilitan el acceso a muchas funciones del sistema operativo y la integración con otras aplicaciones

Programación orientada al objeto

Las metas de la programación orientada al objeto es mejorar la productividad de los programadores haciendo más fácil de rehusar y extender los programas y manejar sus complejidades. De esta forma, se reduce el costo de desarrollo y mantenimiento de los programas. En los lenguajes orientados al objeto los datos son considerados como objetos que a su vez pertenecen a alguna clase. A las operaciones que se definen sobre los objetos son llamados métodos. Ejemplo de programas orientados al objeto: Visual BASIC y C++.

LOGO

LOGO fue diseñado por un educador y matemático del MIT (Instituto de Tecnología de Massachussets), Seymour Papert. Es un lenguaje de programación desarrollado para la enseñanza de los niños; quizá es un lenguaje muy simple, pero es muy poderoso porque incorpora el concepto de procedimientos ayudando a los niños a pensar algorítmicamente (lógica y ordenadamente).

ALGOL

Su nombre se debe a la abreviación de *Algorithmic Language*, es un lenguaje muy poderoso que surgió en la década de 1960. El diseño estuvo a cargo de un comité internacional con sede en Europa y fue el primero de los lenguajes con sintaxis definida de manera formal y matemática por lo cual se considera como el precursor de la familia de lenguajes de programación estructurada, otra de sus características es que el manejo de la memoria es dinámico.

Un programa en ALGOL consta de varios *procedures*, que son módulos que pueden ser llamados recursivamente o estar anidados. Por el tipo de manejo de la memoria, las dimensiones de los arreglos pueden variar durante la ejecución del programa.

ALGOL tiene varias versiones y la mayoría de sus aplicaciones son principalmente científicas. Hacia 1965 dos corrientes se distinguieron sobre el tema de un sucesor para Algol. Como resultado se definieron los lenguajes Algol W que es un lenguaje *minimalista*, rápidamente implementado y distribuido y, por otra parte, Algol 68 que para la época está en la frontera entre un lenguaje para programar en él y un lenguaje para investigar sobre él. Algol W

Lenguaje elaborado diseñado por Niklaus Wirth y Tony Hoare a partir de los trabajos del grupo ALGOL de la IFIP. Se trata de un lenguaje conciso, simple de implementar, que evita todos los defectos conocidos del lenguaje Algol e incluye sus propias características adicionales. Sin embargo, el grupo Algol no lo adoptó como sucesor de Algol prefiriendo en su lugar al que terminó siendo Algol 68. Algol W fue utilizado por gran cantidad de usuarios y sembró el camino para el nacimiento del lenguaje Pascal. Entre las características del lenguaje se destacan: Aritmética de doble precisión, números complejos, Strings y estructuras de datos dinámicas, evaluación por valor, pasaje de parámetros por valor, valor resultado o resultado.

Algol 68

La definición del lenguaje fue presentada en la reunión del comité ALGOL de la IFIP en 1965. Luego de varios años de revisión del diseño se llegó a una versión definitiva en 1968. Al principal autor es Adriaan Van Wijngarden.

Los objetivos principales de ALGOL 68 son el permitir comunicar algoritmos, el permitir una eficiente ejecución de los mismos en diferentes arquitecturas y el de servir como herramienta para la enseñanza.

Una característica interesante de ALGOL 68 es que su semántica fue definida formalmente antes de ser implementado en base al formalismo llamado gramáticas de dos niveles.

EJEMPLO DE PROGRAMA EN ALGOL

```
Procedure Absmax(a) Dimensiones:(n, m) Resultado:(y) Subíndices:(i, k);  
value n, m; array a; integer n, m, i, k; real y;  
comment De la matriz a se toma el elemento con el valor absoluto mayor y se  
coloca en y.  
Los subíndices del elemento se colocan en i y k;  
Begin integer p, q;  
Y: = 0; i: = k: = 1;  
For p: = 1 step 1 until n do  
For q: = 1 step 1 until m do
```

```
If  $abs(a[p, q]) > y$  then  
Begin  $y := abs(a[p, q]);$   
 $i := p; k := q$   
end  
end Absmax
```

JAVA

Java es un lenguaje de programación orientado a objetos desarrollado por James Gosling y sus compañeros de Sun Microsystems al inicio de la década de 1990. A diferencia de los lenguajes de programación convencionales, que generalmente están diseñados para ser compilados a código nativo, Java es compilado en un bytecode que es ejecutado (usando normalmente un compilador JIT), por una máquina virtual Java.

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos mucho más simple y elimina herramientas de bajo nivel como punteros.

Java está sólo lejanamente emparentado con JavaScript, aunque tengan nombres similares y compartan una sintaxis al estilo de C algo parecida.

La plataforma Java y el lenguaje Java empezaron como un proyecto interno de Sun Microsystems en diciembre de 1990. Patrick Naughton, ingeniero de Sun, estaba decepcionado con el estado de C++ y la API de C y sus herramientas. Mientras consideraba migrar a NeXT, Naughton recibió la oferta de trabajar en una nueva tecnología, y así comenzó el proyecto *Stealth*.

El Proyecto Stealth fue rebautizado como *Green Project* (o *Proyecto Verde*) cuando James Gosling y Mike Sheridan se unieron a Naughton. Con la ayuda de otros ingenieros, empezaron a trabajar en una pequeña oficina en Sand Hill Road en Menlo Park, California. Intentaban desarrollar una nueva tecnología para programar la siguiente generación de *dispositivos inteligentes*, en los que Sun veía un campo nuevo a explotar.

El equipo pensó al principio usar C++, pero se descartó por varias razones. Al estar desarrollando un sistema empotrado con recursos limitados, C++ no es adecuado por necesitar mayor potencia además de que su complejidad conduce a errores de desarrollo. La ausencia de un *recolector de basura* (*garbage collector*) obligaba a los programadores a manejar manualmente el sistema de memoria, una tarea peligrosa y proclive a fallos. El equipo también se encontró con problemas por la falta de herramientas portables en cuanto a seguridad, programación distribuida, y programación concurrente. Finalmente abogaban por una plataforma que fuese fácilmente portable a todo tipo de dispositivo.

LÉXICO

Léxico es un *lenguaje* didáctico en español para facilitar el aprendizaje y la enseñanza de la programación orientada a objetos. Pretende, y las investigaciones realizadas con él así lo han comprobado, mejorar el desempeño de los estudiantes y motivarlos para avanzar en la generación de habilidades en lógica de programación pues les permite experimentar con los *algoritmos* diseñados sin tener que dedicar meses a aprender un lenguaje de producción que les permita "ver" sus creaciones.

Su estructura es sencilla y se han retirado elementos de eficiencia innecesarios en un entrenamiento

Permite la suficiente sencillez para entrenarse con los conceptos básicos en algoritmos para que la persona practique las estructuras fundamentales en lógica (secuencia con {...}, decisión con es? y ciclo de repeticiones con mientras) y la clásica estructura de representación de información compuesta llamada arreglo.

El centro medular es la orientación a objetos para lo cual posee el soporte apropiado y se ha simplificado al máximo de manera que ayude a la inmersión en los conceptos.

Los objetos pueden ser construidos con base en las dos clases fundamentales, cantidad y caracteres, con base en las clases establecidas por el programador y con base en las 7.000 definiciones de la plataforma .net de Microsoft, lo que permite desarrollar verdaderas aplicaciones que incluyan controles y el manejo de eventos.

POSTSCRIPT

En informática, lenguaje de descripción de páginas de Adobe Systems que ofrece diferentes posibilidades de fuentes así como gráficos de alta calidad. El lenguaje PostScript utiliza comandos parecidos al inglés para controlar el diseño de las páginas y para cargar y escalar fuentes de contorno. Dado que el PostScript utiliza fuentes escalables de contorno, de forma muy parecida a los gráficos orientados a objetos, puede crear una fuente de cualquier tamaño, permitiendo al usuario una mayor flexibilidad al crear documentos. Gran cantidad de impresoras utilizan el lenguaje PostScript, bien como único modo de impresión o como alternativa a otros sistemas. Adobe también produce Display PostScript, un lenguaje de gráficos para pantallas de computadora que permite un WYSIWYG total al usuario de PostScript y Display PostScript, lo cual es difícil cuando se utilizan métodos distintos para la pantalla y la impresora.

PERL

Perl, Lenguaje Práctico para la Extracción e Informe (ver abajo) es un lenguaje de programación diseñado por Larry Wall creado en 1987. Perl toma características del C, del lenguaje interpretado shell (sh), AWK, sed, Lisp y, en un grado inferior, muchos otros lenguajes de programación. Perl es un lenguaje de propósito general originalmente desarrollado para la manipulación de texto y que ahora es utilizado para un amplio rango de tareas incluyendo administración de sistemas, desarrollo web, programación en red, desarrollo de GUI y más.

Se previó que fuera práctico (facilidad de uso, eficiente, completo) en lugar de hermoso (pequeño, elegante, mínimo). Sus principales características son que es fácil de usar, soporta tanto la programación estructurada como la programación orientada a objetos y la programación funcional, tiene incorporado un poderoso sistema de procesamiento de texto y una enorme colección de módulos disponibles

Perl tiene muchas y variadas aplicaciones, gracias a la disponibilidad de muchos módulos estándares y de terceras partes.

Se ha usado desde los primeros días del Web para escribir guiones (*scripts*) CGI y es un componente integral de la popular solución LAMP para el desarrollo web. Grandes proyectos escritos en Perl son Slash, IMDb [1] primeras implementaciones de PHP [2] Perl se usa a menudo como un "lenguaje pegamento" ^(en), uniendo juntos sistemas e interfaces que no fueron específicamente diseñadas para interoperar; y para el "escarbado de datos", convirtiendo o procesando grandes cantidades de datos para tareas como por ejemplo crear informes. De hecho, estas fortalezas están íntimamente unidas. Su combinación hace a Perl una popular herramienta de propósito general para los administradores de sistemas, especialmente en programas pequeños que pueden ser escritos y ejecutados en una sola línea de comandos. Perl es también ampliamente usado en finanzas y bioinformática, donde es apreciado por su desarrollo rápido, tanto de aplicaciones como de despliegue, así como la habilidad de manejar grandes volúmenes de datos.

FORTH

Lenguaje de programación para computadores ideado por Charles Moore y Elisabeth Rather entre los años 1965 y 1970 en el National Radio Astronomy Observatory de Kitt Peak, Arizona.

Inicialmente diseñado para una aplicación muy concreta: la astronomía (cálculo de trayectorias de cuerpos en órbita, cromatografías, análisis de espectros de emisión), ha evolucionado hasta ser aplicable a casi todos demás campos relacionados o no con esa rama de la ciencia (cálculos de probabilidad, bases de datos, análisis estadísticos y hasta financieros).

Su nombre es una contracción de la palabra inglesa fourth, dado que sus creadores le consideraban destinado a cuarta generación de computadoras, pero la primera edición del lenguaje fue preparada para un IBM 1130, que solo permitía nombres con una longitud máxima de cinco letras; su nombre se quedó ya para siempre en FORTH.

Posteriormente, un programa para la adquisición automática y continua de datos realizado en este lenguaje ha descubierto al menos la mitad de los cúmulos interestelares conocidos en la actualidad.

PROLOG (Programming in Logic)

Es un lenguaje de programación interpretado que fue inventado en 1973 por Alain Colmerauer. Se utiliza en programas de inteligencia artificial.

Es un programa conocido de quinta generación. Japón lo ha adoptado como el lenguaje oficial para la inteligencia artificial. Permite que a través de una regla se puedan obtener hechos o inferencias que permitan ubicar deducciones lógicas.

Prolog es un lenguaje declarativo y funcional. Carece de las declaraciones que se pueden observar en un lenguaje imperativo, como pueden ser las declaraciones de tipos, de variables, de procedimientos, etc.

HASKELL

Haskell es un lenguaje de programación puramente funcional de propósito general. Su nombre proviene del lógico Haskell Curry.

En los años 1980 se constituyó un comité cuyo objetivo era crear un lenguaje funcional que reuniera las características de los múltiples lenguajes funcionales de la época, como Miranda, y resolviera la confusión creada por los múltiples lenguajes que seguían este paradigma. El lenguaje evoluciona rápidamente con y (ver más abajo) como los representantes actuales del estándar *de facto*. El último estándar *semi-oficial* es **Haskell 98**, con la intención de especificar una versión mínima y compatible del lenguaje como base para futuras extensiones y para su enseñanza.

Las características más interesantes de Haskell incluyen el soporte para tipos de datos y funciones recursivas, listas, tuplas, guardas y reconocimiento de patrones. La combinación de las mismas puede resultar en algunas funciones casi triviales cuya versión en lenguajes imperativos pueden llegar a resultar extremadamente tediosas de programar. Haskell es, desde 2002, uno de los lenguajes funcionales sobre los que más se ha investigado.

MODULA-2

Modula-2 es un lenguaje de programación cuyo autor es Niklaus Wirth, autor también del lenguaje Pascal.

Como novedad respecto a este último lenguaje, introduce el concepto de módulo, y de encapsulación. Del código contenido en un módulo, sólo se facilita una interfaz pública, permaneciendo el resto oculto para un desarrollador ajeno (encapsulado), lo que facilita el mantenimiento de dichas estructuras de programación.

Este concepto de módulo constituye el antecedente de las clases u objetos que se observan en el concepto moderno de Programación Orientada a Objetos (POO); sin embargo, la incapacidad de declarar múltiples instancias de los módulos, así como la ausencia de todo tipo de herencia, impiden decir que Modula-2 sea un lenguaje orientado a objetos propiamente tal.

Modula-2 se utiliza principalmente en las universidades por su excelente adaptación a la enseñanza de lenguajes estructurados.

Existen compiladores de Modula-2 como Mocka (para Linux) o FST (para MS-DOS) por citar sólo dos ejemplos.

PYTHON

Es un lenguaje de programación habitualmente comparado a TCL, Perl, Scheme, Java y Ruby. Actualmente, Python se desarrolla como un proyecto de código abierto, administrado por la Python Software Foundation. La última versión estable del lenguaje es actualmente (Septiembre de 2006) la 2.5. Guido van Rossum, más conocido como Guido, creó Python, un lenguaje de programación de scripting, la "oposición leal" a Perl, lenguaje con el cual mantiene una rivalidad amistosa. Los usuarios de Python consideran a éste mucho más limpio y elegante para programar.

Python permite dividir el programa en módulos reutilizables desde otros programas Python. Viene con una gran colección de módulos estándar que se pueden utilizar como base de los programas (o como ejemplos para empezar a aprender Python). También hay módulos incluidos que proporcionan E/S de ficheros, llamadas al sistema, sockets y hasta interfaces a GUI (interfaz gráfica con el usuario) como Tk, GTK, Qt entre otros...

Python es un lenguaje interpretado, lo que ahorra un tiempo considerable en el desarrollo del programa, pues no es necesario compilar ni enlazar. El intérprete se puede utilizar de modo interactivo, lo que facilita experimentar con características del lenguaje, escribir programas desechables o probar funciones durante el desarrollo del programa. También es una calculadora muy útil.

El nombre del lenguaje proviene de la afición de su creador original, Guido van Rossum, por los geniales humoristas británicos Monty Python. El principal objetivo que

persigue este lenguaje es la facilidad, tanto de lectura, como de diseño. Python es un lenguaje de programación multiparadigma. Esto significa que más que forzar a los programadores a adoptar un estilo particular de programación, permite varios estilos: Programación orientada a objetos, programación estructurada, programación funcional y programación orientada a aspectos. Otros muchos paradigmas más están soportados mediante el uso de extensiones. Python usa tipado dinámico de datos y reference counting para el manejo de memoria. Una característica importante del Python es la resolución dinámica de nombres, lo que enlaza un método y un nombre de variable durante la ejecución del programa.

Otro objetivo del diseño del lenguaje era la facilidad de extensión. Nuevos módulos pueden ser fácilmente escritos en C o C++. Python puede ser utilizado como un lenguaje de extensión para módulos y aplicaciones que necesitan de una interfaz programable. Aunque el diseño de Python es de alguna manera hostil a la programación funcional tradicional del Lisp, existen bastantes analogías entre Python y los lenguajes minimalistas de la familia del Lisp como puede ser el Scheme.

PL/SQL

Lenguaje de programación de Oracle.

El PL/SQL soporta todas las consultas y manipulación de datos que se usan en SQL, pero incluye nuevas características:

El manejo de variables.

Estructuras modulares.

Estructuras de control de flujo y toma de decisiones.

Control de excepciones.

El lenguaje PL/SQL está incorporado en:

Servidor de la base de datos.

Herramientas de Oracle (Forms, Reports,...).

En un entorno de base de datos los programadores pueden construir bloques PL/SQL para utilizarlos como procedimientos o funciones, o bien pueden escribir estos bloques como parte de scripts SQL*Plus.

Los programas o paquetes de PL/SQL se pueden almacenar en la base de datos como otro objeto, y todos los usuarios que estén autorizados tienen acceso a estos paquetes. Los programas se ejecutan en el servidor para ahorrar recursos a los clientes.

PHP

PHP es un lenguaje de programación usado generalmente para la creación de contenido para sitios Web. Las siglas significan "**PHP Hypertext Pre-processor**" (inicialmente PHP Tools, o, *Personal Home Page Tools*), y se trata de un lenguaje interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios Web. Últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica usando la librería GTK+.

Debido al diseño de PHP, también es posible crear aplicaciones con una interfaz gráfica para el usuario (también llamada GUI), utilizando la extensión PHP-GTK. También puede ser usado desde la línea de órdenes, de la misma manera como Perl o Python pueden hacerlo, esta versión de PHP se llama PHP CLI (*Command Line Interface*).

Su interpretación y ejecución se da en el servidor, en el cual se encuentra almacenado el script, y el cliente sólo recibe el resultado de la ejecución. Cuando el cliente hace una petición al servidor para que le envíe una página web, generada por un script PHP, el servidor ejecuta el intérprete de PHP, el cual procesa el script solicitado que generará el contenido de manera dinámica, pudiendo modificar el contenido a enviar, y regresa el resultado al servidor, el cual se encarga de regresárselo al cliente. Además es posible utilizar PHP para generar archivos PDF, Flash, así como imágenes en diferentes formatos, entre otras cosas.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, Postgres, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite; lo cual permite la creación de Aplicaciones web muy robustas.

PHP también tiene la capacidad de ser ejecutado en la mayoría de los sistemas operativos tales como UNIX (y de ese tipo, como Linux), Windows y Mac OS X, y puede interactuar con los servidores de web más populares ya que existe en versión CGI, módulo para Apache, e ISAPI.

El modelo PHP puede ser visto como una alternativa al sistema de Microsoft que utiliza ASP.NET/C#/VB.NET, a ColdFusion de la compañía Macromedia, a JSP/Java de Sun Microsystems, y al famoso CGI/Perl. Aunque su creación y desarrollo se da en el ámbito de los sistemas libres, bajo la licencia GNU, existe además un IDE comercial llamado Zend Optimizer.

Bibliografía

www.wikipedia.com

www.monografias.com

http://entrenate.dgsca.unam.mx/introduccion/leng_alton.html