SÍNTESIS

El presente trabajo tiene como objetivo el presentar una alternativa de solución para la actualización de algunas materias contenidas en el plan de estudios de la carrera de Ingeniería en Sistemas Computacionales del Instituto Tecnológico de Nuevo León, y sugerir la inclusión de otras materias nuevas. El énfasis se centra en la línea de la enseñanza de la programación de computadoras, ya que dicha área es la puerta de entrada a la especialidad de la carrera, El presente trabajo incluye los siguientes puntos:

- Mención del marco histórico y actual de los lenguajes y ambientes de desarrollo de programas para computadoras, y la importancia de la enseñanza de métodos algorítmicos para el diseño de soluciones computacionales.
- Analizar los diferentes lenguajes de programación de computadoras existentes en el mercado para seleccionar el mas adecuado en base a ciertos criterios definidos, de forma que los alumnos aprendan a desarrollar aplicaciones con las herramientas, métodos y técnicas de actualidad.
- La determinación de un segundo lenguaje que facilite la construcción y uso de Bases de datos, de forma similar a como se hace en un ambiente laboral.
- Analizar la naciente área de desarrollo de aplicaciones en ambiente Internet/Intranet, los lenguajes mas usados en ese tipo de desarrollos, y sugerir respuesta a esta nueva necesidad.
- Análisis de la situación actual del plan de estudios y la falta de materias que muestren a los alumnos los conceptos y lenguaje de la función de negocios.

En el capitulo de conclusiones y recomendaciones se reflejan las sugerencias de respuesta a los problemas planteados.

1

INTRODUCCIÓN

A manera de introducción es conveniente mencionar que existen dos tipos de computadoras cuyas aplicaciones son muy diferentes: computadoras digitales y computadoras analógicas, el tipo de información que operan también es diferente, ya que la computadora analógica opera con información de magnitudes físicas (temperatura, presión, voltaje, etc.), su tratamiento y operación quedan fuera del alcance de este escrito.

Las computadoras digitales son máquinas que en su funcionamiento usan el sistema numérico binario para representar las instrucciones de los programas y de los datos que se van a procesar. A este respecto la electrónica que se usa para su construcción es la electrónica de flip-flop que facilita la construcción de dispositivos que distinguen la situación entre 2 estados posibles: apagado y encendido(ó "0 y 1"), el presente trabajo se relaciona con la programación de las computadoras digitales.

La programación de computadores digitales es una especialidad profesional muy importante en el área de los Sistemas Computacionales, y quien desea dedicar su tiempo a esta actividad debe obligatoriamente estar actualizado en cuanto a métodos y técnicas modernas de desarrollo para crear aplicaciones ó programas usando la mas moderna tecnología.

1.1 Descripción del problema

La tecnología computacional evoluciona rápidamente y en esa evolución se diseñan nuevos dispositivos, métodos y técnicas para el correcto aprovechamiento de las características de las innovaciones tecnológicas generadas.

La utilización de las innovaciones tecnológicas de parte de los diferentes sectores del ámbito laboral es casi inmediata por la ventaja competitiva que les presenta a sus negocios.

Ante este panorama el Instituto Tecnológico de Nuevo León como todas las Instituciones de educación necesita adecuar constantemente los planes de estudio de las asignaturas de sus carreras profesionales, incorporando la enseñanza de las nuevas técnicas de computación que van surgiendo, con el objetivo de que sus alumnos dispongan de conocimiento actualizado a la par con los requerimientos de los diferentes sectores del ámbito laboral.

De no hacer lo anterior se tendría el peligro que sus alumnos no dispongan del conocimiento de los dispositivos, métodos y técnicas de desarrollo actualizadas y no podrían dar respuesta a las necesidades del sector laboral en la utilización de las mas modernas tecnologías.

1.2 Objetivo.

El objetivo de esta Tesis es presentar una propuesta al Instituto Tecnológico de Nuevo León en cuanto a la enseñanza de la programación de computadoras digitales a los alumnos de la carrera de Ingeniería en Sistemas Computacionales, este aprendizaje les es dado en los primeros semestres de la carrera y representa la inducción básica del alumno en el ambiente propio de la profesión que ha elegido, y su ejercicio será la base para el desarrollo de su vida profesional.

En el transcurso de los últimos 10 años (1990-1999) se han desarrollado y puestos en el mercado de la computación un gran numero de nuevos lenguajes de programación, así como nuevas versiones de lenguajes que ya se usaban, éstos nuevos lenguajes han incorporado nuevas técnicas de desarrollo de programas tales como : programación orientada a objetos, programación orientada a eventos, también a los lenguajes les han implementado herramientas de ambiente visual para el Desarrollo Rápido de Aplicaciones (RAD).

Actualmente en el Instituto tecnológico de Nuevo León, en las materias cuya función es la de la enseñanza de la programación de computadoras, se enseña el Lenguaje C++, en su versión como lenguaje de tercera generación.

1.3 Hipótesis.

Ante la evolución de los lenguajes de programación, y el surgimiento de nuevos métodos y técnicas para la gestión de la información, se pretende demostrar la urgente necesidad de modificar el plan de estudios de la carrera de Ingeniería en Sistemas Computacionales, para adecuarla a los nuevos tiempos de retos y competencia.

Asimismo de la necesidad de incluir asignaturas ó temas que complementen el perfil del egresado para la competencia administrativa, esto además puede facilitar que en el aprendizaje de las técnicas de programación se realicen talleres ó trabajos de laboratorio que conjuguen los métodos administrativos y de producción con las técnicas de desarrollo de aplicaciones computacionales.

1.4 Limites del estudio.

Las fronteras del presente estudio serán el marco de referencia de las metas y objetivos del Instituto tecnológico de Nuevo León.

1.5 Justificación.

El ámbito de mercados globales al cual ha ingresado nuestro país, ha generado que el sector laboral demande profesionistas en el área de Sistemas Computacionales capacitados para desarrollarse en estas nuevas situaciones, donde se requiere que el nuevo profesionista tenga:

- Conocimiento de las funciones básicas de los negocios.
- Conocimientos actualizados de técnicas del desarrollo de aplicaciones usando la mas moderna tecnología computacional.
- Valores humanos tales como la honestidad, responsabilidad, respeto, trato amable, etc...
- Dominio de un segundo idioma.

Las características mencionadas son referidas como rasgos del perfil de los nuevos profesionistas, y han sido expresadas por expertos conocedores de estos temas, tanto en foros y conferencias como también en la prensa escrita. En el presente trabajo de Tesis solo se tiene alcance respecto a los dos primeros puntos.

1.6 Metodología a seguir.

La metodología a utilizar será la recolección de los datos para la sustentación de la Tesis, y se define mediante los siguientes puntos:

- Análisis de herramientas y ambientes de desarrollo usados actualmente en los diferentes sectores del ámbito laboral de la localidad.
- Análisis de las características técnicas de los lenguajes de programación y sus requerimientos de operación.
- Análisis de plataformas de operación.
- Selección de lenguajes Tipo
- Definición de la secuencia de los temas para la enseñanza de los lenguajes de programación y las materias donde se incluirán.
- Definición de los procesos administrativos básicos a incluir en la enseñanza, para la capacitación de los alumnos en ellos.
- Capacitación de la planta docente de la especialidad de Sistemas Computacionales.

1.7 Revisión Bibliográfica.

Para los temas que refieren los antecedentes cronológicos de las generaciones de la computadora se consideró el texto **Elementos de Computación** de Guillermo Levine, de la Editorial Mc Graw Hill.

En lo referente a la Clasificación de los Lenguajes de Programación se consideraron tres textos a saber: **Curso de Programación** de cuyos autores son Jorge Castro, Felipe Cucker, Xavier Messenguer, Albert Rubio y Luis Solano de la editorial Mc Graw Hill, **Lenguajes de Programación** de Allen B. Tucker también de la editorial Mc Graw Hill, **Intelligent Software Systems Develpment an IS Manager's Guide** de Paul Harmon & Curtis Hall de la editorial Jhon Wiley & Sons.

Del tema Panorama histórico de los Lenguajes de Programación se consultó al libro **Programación Orientada a Objetos, Conceptos, Modelado, Diseño y Codificación en C++** del autor **Luis Joyanes Aguilar** , asimismo al texto **Intelligent Software Systems Develpment an IS Manager's Guide,** y el **fascículo No. 18** de la serie **Curso IBM de Programación.**

Para el tema del Primer lenguaje de Programación en lo referente a las características de los nuevos lenguajes de programación así como sus métodos y técnicas de programación, se consultaron los **fascículos 1 y 2 de la serie Curso IBM de Programación**, así como algunas páginas de Internet por ejemplo :

http://members.es.tripod.de/adm/popup/roadmap-search.html

http://www.recursos-as400.com

http://www.audisoft.com

http://www.programacion.net/cursos/vbcliserv

http://www.concytec.gob.pe/institucional/servidor.htm

http://www.angel.perez.net/MODINET/desarr.htm

http://www.terabyte.cl/cobolwow.htm

2

Evolución y clasificación de los lenguajes de programación.

2.1 Generaciones de la Computadora.

Históricamente las computadoras digitales han sufrido una evolución que puede relacionarse en términos de "generaciones" de las cuales algunos autores de textos las definen en tres y media(ó casi cuatro), a saber:

Primera generación : Aparece a fines de la década de 1940 y termina a inicios de 1960 la característica que la distingue es su tecnología, ya que fueron construidas a base de bulbos ó tubos de vacío, debido a ello ocupan grandes espacios de los edificios, y su programación representaba una gran dificultad ya que solo se podían programar en lenguaje de máquina. Algunos de los modelos de esta primera generación son:

ENIAC. (1947) Fue la primera computadora digital en la historia, fué un prototipo experimental , no era programable, fue construida en el sótano de la universidad de Pennsylvania de los Estados Unidos de America por los doctores Jhon W. Mauchly y J. Presper Eckert.

EDVAC. (1949) También fue solo un prototipo de laboratorio pero incluía el concepto de "Programa almacenado" tal como las computadoras actuales, el concepto fue acuñado por el doctor Jhonn Von Neumann.

UNIVAC I. (1951) Desarrollada por los doctores Mauchly y Eckert, que fundaron la empresa UNIVAC, fue la primera computadora comercializada, el primer cliente en adquirir este modelo fue la oficina del censo de los Estados Unidos de América.

IBM 701.(1953) La compañía IBM fundada por Thomas Watson, se dedicaba a comercializar equipo de registro unitario, el cual usaba tarjetas perforadas como el medio de alimentación y almacenamiento de datos, y su reconocimiento se conseguía mediante la interpretación magnética del código desarrollado por Herman Hollerit en 1890. Fue el primer modelo comercializado por esta compañía . que llegaría a una posición dominante en años posteriores.

1954 – 1960. Fue un periodo donde IBM presentó varios modelos que utilizaban un medio de almacenamiento llamado "Tambor magnético" con lo que marcó el liderazgo en el ámbito computacional, posteriormente implementó el uso de discos magnéticos llamados disk pack. En este tiempo también aparece la compañía Remington Rand y la UNIVAC continuó presentando nuevos modelos.

Segunda generación : (1960-1964) A principios de la década de 1960 la industria japonesa desarrolla aplicaciones del transistor en la electrónica, y su primera aplicación es en la producción de radios de transistores, los fabricantes de computadoras no tardan en darse cuenta que el transistor puede también emplearse con las computadoras.

Asimismo también aparecen los primeros lenguajes de "alto nivel" y con ello el "oficio" de programador de computadoras.

De esta manera las características de las computadoras de la segunda generación son las siguientes:

- Construidas en base a electrónica de transistores.
- Inicia la programación simboloca en base a lenguajes de alto nivel(Cobol, Fortran).
- Aparecen nuevas compañías fabricantes de computadoras

Algunos de los nuevos modelos que podemos citar son:

CDC 3000. Aparece la compañía CDC(Control data Company), el modelo se caracterizaba por ser muy una computadora grande, potente y veloz en sus operaciones.

BURROUGHS 5000. Nace la compañía Burroughs, y su serie 5000 es considerada la mas avanzada de la época.

UNIVAC siguió ofreciendo nuevos modelos de computadoras.

IBM. siguió ofreciendo nuevos modelos mejorados cada vez .

Tercera generación : (1964-1970) Esta generación fue marcada por la empresa IBM al presentar en el año de 1964 un nuevo concepto de atención al mercado, anunciando la llamada "Familia 360" que ofrecía la herramienta para la solución de problemas de negocios ya fueran pequeños, medianos ó grandes, pues ofrecía modelos al tamaño del cliente. Asimismo la tecnología de transistores evolucionó - a la tecnología de circuitos integrados mediante la miniaturiza- - ción de los transistores, logrando que las computadoras disminu- - yeran significativamente su tamaño, pero aún seguían ocupando grandes áreas que debían estar acondicionadas especialmente en cuanto a humedad, aire acondicionado, grandes unidades de res- - paldo de energía, y equipo de regulación de voltaje. Respecto del software se tiene también un avance ya que aparecen los primeros sistemas operativos que permiten una comunicación entre la computadora y el programador.

Sus características que la distinguen son las siguientes:

- Construidas en base a electrónica de circuitos integrados
- Aparecen los primeros Sistemas Operativos

Las compañías fabricantes continúan generando nuevos modelos:

IBM. Ofrece su "Familia" de computadoras "Serie 360", a inicios de la década de 1970 evoluciona a Familia de la "Serie 370", y también ofrece mejoras con un nuevo Sistema Operativo.

CDC Ofrece el modelo CDC serie 6000, destacándose como el fabricante de las computadoras mas grandes y veloces, a inicios de la década de 1970 ofrece el modelo CDC serie 7000, que después fueron substituidos por la serie CDC Cyber.

BURROUGHS Ofrece el modelo Burroughs serie 7000, y un Sistema Operativo muy robusto llamado MCP.

UNIVAC Ofrece la Familia 1100.

A principios de 1970 aparecieron computadoras que ya no eran grandes aparatos ocupando grandes áreas especialmente acondicionadas, sino que eran computadoras de mediano tamaño que podían ser instaladas en lugares reducidos y sin acondicionamiento especial. A estas computadoras se les llamó "Minicomputadoras" y fueron desarrolladas por compañias que recién ingresaban al mercado tales como la Digital, Hewlett Packard, Data General, Wang, y otras mas. Un ejemplo de ello fueron:

DIGITAL Su primer minicomputadora de éxito comercial fue la PDP-8. La compañía fue llamada Digital Equipment Corporation quien después presentó otro modelo denomi.nado PDP-11, siguiéndole después con un nuevo modelo mejorado de gran aceptación, el modelo VAX.

HEWLETT PACKARD empresa que entró al mercado de las minicomputadoras con el modelo de la serie HP-3000, posteriormente introdujo un nuevo mode-

lo de la serie HP-9000.

IBM Reforzó su presencia en el mercado de las mini-computadoras con los modelos IBM-34, IBM-36, IBM-38.

Cuasi Cuarta generación. Esta generación fue marcada con la aparición de las microcomputadoras que utilizando tecnología VLSI, integración de circuitos electrónicos a gran escala revolucionó grandemente el mercado de las computadoras, encontrándole aplicaciones en todos los ámbitos: en el hogar, personal, y de negocios .

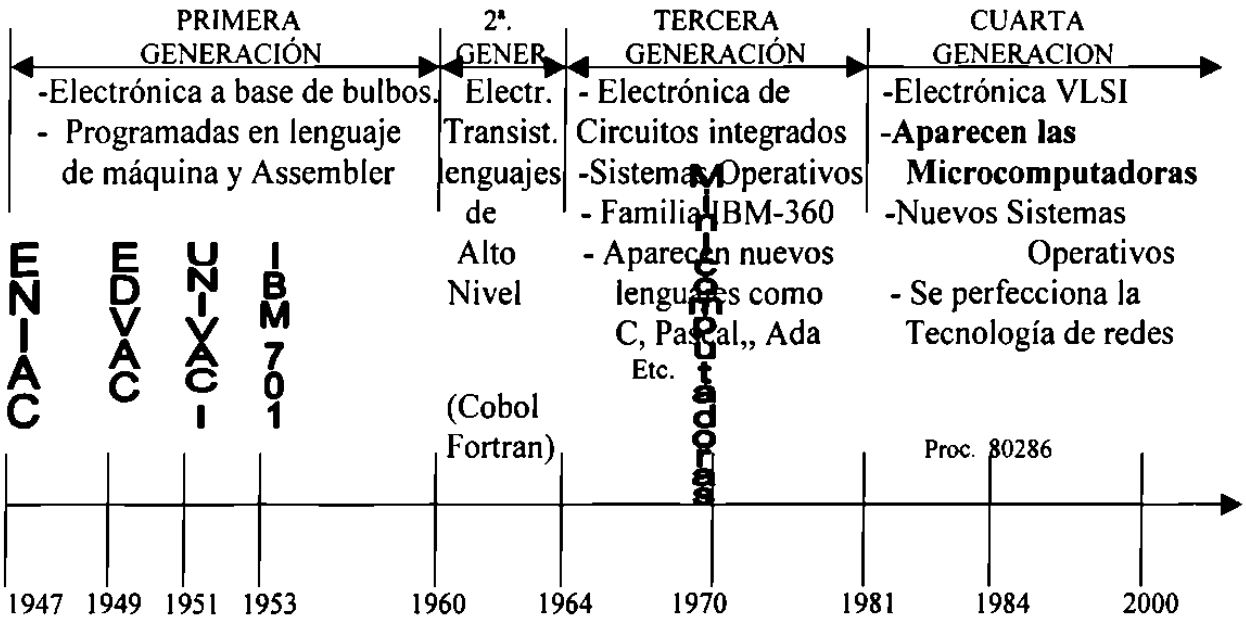


Fig. 1 Cronograma de las generaciones de las computadoras

2.2 Clasificación de las Computadoras.

De acuerdo a su tamaño las computadoras se clasifican en:

- Supercomputadoras
- Macrocomputadoras
- Minicomputadoras
- Microcomputadoras(ó PC's)

Supercomputadoras

La Supercomputadora es el tipo de computadora mas potente y mas rápida en la ejecución de operaciones que puede existir en un tiempo dado, son equipos de gran tamaño y ocupan espacios grandes. Estas máquinas son diseñadas para procesar grandes cantidades de información en poco tiempo, y son dedicadas a tareas específicas. Asimismo son el tipo de computadoras mas caras ya que son valuadas a precios de millones de dólares, una de sus características es que deben estar instaladas en lugares debidamente acondicionados para control especial de temperaturas, para disipar el calor que se produce en algunos de sus componentes. Algunas de sus aplicaciones son:

1 Búsqueda y estudio de energías y armas nucleares.
2 Búsqueda de yacimientos petrolíferos con grandes bases de datos.
3 Estudio y predicción de tornados.
4 Estudio y predicción de climas de cualquier parte del mundo.
5 Elaboración de maquetas y proyectos de creación de aviones, simuladores de vuelo, etc..

debido a que son computadoras muy caras se construyen pocas de ellas al año.

Macrocomputadoras.

Estas computadoras también son llamadas "Mainframe", y también como las supercomputadoras ocupan grandes espacios, realizan operaciones ejecutables también a grandes velocidades, su diseño va dirigido a tener conexión y controlar a cientos de usuarios simultáneamente, así como con cientos de dispositivos de entrada y salida de datos. Los Maiframes son computadoras algunas veces mas caras que las supercomputadoras, y algunas veces también con mayor capacidad de servicio, pero las supercomputadoras siempre son mas rápidas en cuanto a ejecución de programas.

En el pasado los Maiframes ocupaban salas de computo completas y algunas veces hasta pisos completos de edificios, pero actualmente los Mainframes son parecidos a una fila de archiveros en un cuarto con piso falso, ya que esto así se diseña para ocultar bajo piso los cientos de cables de conexión a periféricos. La temperatura del Site debe ser controlada.

Minicomputadoras.

En 1970 aparece la Minicomputadora, una versión mas pequeña de la Macrocomputadora, desde su aparición fue dirigida a tareas específicas y solo se conectaba solo a los periféricos mas elementales, lo cual se vió reflejado en reducción de precio y costos de mantenimiento. En cuanto a su capacidad de cómputo las Minicomputadoras se encuentran entre los Maiframe y las estaciones de trabajo.

En general una Minicomputadora es un sistema de multiproceso(con capacidad de ejecutar procesos paralelos), puede soportar de 10 hasta 200 usuarios simultáneamente, actualmente se usan para almacenar grandes bases de datos, en la automatización industrial y aplicaciones multiusuario.

Microcomputadoras.

Las Microcomputadoras tuvieron su origen con el surgimiento de los microprocesadores, un microprocesador es "una computadora en un chip", ó sea en un circuito integrado independiente. Las Minicomputadoras ó PC's son computadoras de uso personal y relativamente baratas, actualmente son muy usadas en oficinas, escuelas y los hogares.

El término PC fue originado en el año de 1981 cuando IBM sacó al mercado su modelo "IBM PC", el cual se convirtió en un tipo de computadora ideal para uso "personal", de ahí ese término cobró popularidad. Posteriormente otras empresas entraron al mercado de las PC's usando procesadores del mismo tipo que las de IBM y las llamaron "PC compatibles" con IBM. Existe otro tipo de microcomputadoras que no son compatibles con IBM, la cual es la marca Macintosh pero que también se les llama "PC" por ser de uso personal.

2.3 Clasificación del software.

De todos los inmersos en el ámbito de los Sistemas Computacionales(también llamado ámbito de la Informática, y de la Computación), es sabido que una computadora digital se concibe en forma general bajo dos conceptos a saber: Hardware y Software.

El hardware se refiere a todos los componentes físicos que forman la máquina que llamamos computadora(Gabinete, Monitor, Tarjetas de circuitos impresos, cables, etc..), en suma son "todas aquellas partes que podemos ver y tocar".

El software se refiere a los elementos intangibles ó sea aquellas partes que no podemos tocar(los programas), pero que al ser "integradas" al hardware se logra que la maquinaria inerte "adquiera vida artificial" y pueda realizar funciones de procesamiento de datos. Metafóricamente el software para el hardware es como el combustible para el automóvil, un elemento que no es parte integral de él, pero que le es indispensable para su funcionamiento.

El software ha sido un pilar para la utilización efectiva del hardware, y los programadores son los artífices "creadores", sus herramientas principales son :

- Los lenguajes de programación(creados por ellos mismos), donde se definen los métodos y técnicas de la programación.
- Su propia imaginación.

para lograr el desarrollo de aplicaciones que sean soluciones a los diferentes tipos de problemas, a través de darle "ordenes" a la computadora para que ejecute las tareas deseadas.

Al igual que el hardware el software ha evolucionado a través del tiempo, y por él se ha logrado que el ser humano "se comunique" con la computadora y le pueda indicar que realice actividades ó tareas definiéndole claramente qué hacer, como hacerlo y cuando hacerlo.

Por medio de los lenguajes de programación se hace uso de las características del hardware y mediante el acceso a la memoria de la máquina se deja (almacena) en ella los datos con los que trabajará y las instrucciones que le indiquen el procedimiento a seguir para la realización de la tarea deseada. En forma general el software se clasifica en dos tipos:

- Software del Sistema(de sistemas, de tiempo real).
- Software de Aplicación(de gestión, de ingenieria y científico, empotrado, de PC, de IA).

2.3.1 Software de Sistemas.

El software de sistemas es un conjunto de programas diseñados para facilitar la operación de otros programas(compiladores, editores, utilerías de gestión de archivos, etc..), los cuales tienen capacidad de operar con estructuras de información.

Otras aplicaciones tales como Sistemas Operativos, utilerías para la operación de periféricos y de procesadores de telecomunicaciones permiten procesar grandes volúmenes de datos.

El software de sistemas se caracteriza por operar en gran manera con el hardware de la computadora, permitir el acceso a múltiples usuarios, operación de procesos concurrentes, planificación de uso y comparición de recursos, optima planeación de procesos, y conexión a múltiples interfaces externas.

Como software de tiempo real es conocido aquel software que permite medir, analizar y controlar sucesos del mundo real conforme estos ocurren, los elementos de software de tiempo real incluyen un componente de acumulación de datos que recolecta y formatea la información en forma externa, también un componente de análisis que transforma la información según lo requiere la aplicación. El software de tiempo real también tiene un componente de monotorización que coordina a todos los demás componentes, de forma que pueda mantenerse la respuesta en tiempo real(generalmente en el rango de 1 milisegundo a un minuto).

Es conveniente precisar que el término "tiempo real" es diferente del término "tiempo compartido" ya que el un sistema de tiempo compartido puede sobrepasarse el tiempo sin que se ocasione un contratiempo.

2.3.2 Software de Aplicación.

Este tipo de software implementa los procedimientos requeridos para realizar las funciones del procesamiento de la información.

Software de gestión.

El procesamiento de información comercial constituye la mayor de las áreas de aplicación del software, los sistemas administrativos(Nóminas, Contabilidad, Control de inventarios, etc..) son considerados dentro del software de gestión, ya que opera con una ó mas Bases de datos donde se almacenan los datos de la información comercial.

Las aplicaciones de esta área reestructuran los datos existentes para facilitar las operaciones comerciales y la toma de decisiones. Además de las tareas convencionales de procesamiento de datos también realizan operaciones interactivas, por ejemplo transacciones en puntos de venta.

Software de ingeniería y científico.

El software de ingeniería y científico se caracteriza por los algoritmos de "Métodos Numéricos", estas aplicaciones van desde la astronomía a la vulcanología, desde el análisis de la presión de automotores a la dinámica orbital de las lanzaderas espaciales y desde la biología molecular a la robótica. Sin embargo las nuevas aplicaciones de esta área se han alejado de los métodos convencionales. El diseño asistido por computadora(CAD), la simulación de sistemas y otras aplicaciones interactivas han comenzado a tomar características de software de tiempo real.

Software empotrado.

Este software es definido de esta manera porque reside en "Memoria de solo lectura(ROM)", y se utiliza para controlar productos y sistemas de los mercados industriales y de consumidores. Este software puede ejecutar funciones muy limitadas(por ejemplo el control de las teclas de un horno de microondas) para facilitar una función significativa y capacidad de control, tales como: control digital de la gasolina de un automotor, sistemas de frenado, etc..

Software de Computadoras Personales (PC).

Este software ha desarrollado grandemente su potencial desde la década pasada, y algunas de sus aplicaciones son: Procesamiento de textos, hojas de cálculo, graficos por computadora, entretenimiento, gestión de bases de datos, aplicaciones financieras, comerciales y personales, de redes externas y acceso a bases de datos. Este tipo de software representa una de la áreas de mayor innovación.

Software de IA(Inteligencia Artificial).

Este software hace uso de algoritmos heurísticos(no numéricos), y se usa para el diseño de soluciones a problemas complejos que no se pueden resolver directamente con cálculo numérico ó análisis directo. Esta área es tambien llamada "de los Sistemas Expertos" ó "Sistemas basados en conocimiento".

2.4 Evolución y clasificación de los lenguajes de programación.

Para la solución de problemas el hombre ha diseñado métodos de acuerdo a las herramientas de que dispone, con la aparición de la computadora y de su adopción como herramienta de solución a cierto tipo de problemas, el hombre ha diseñado también métodos acordes al uso de esta herramienta, y los ha adaptado a la codificación de los lenguajes de programación.

A muchos problemas que son tratados mediante la computadora el medio externo les proporciona cierto dinamismo, y en esos casos los métodos también necesitan modificarse como adecuación a los medios externos, de la misma forma los lenguajes de programación han evolucionado para adecuarse a las nuevas circunstancias.

Los métodos y lenguajes de programación han evolucionado rápidamente desde los primeros lenguajes de alto nivel que aparecieron en la década de 1950. En seguida se enuncian los factores mas importantes que han influido en esta evolución.

El hardware y los Sistemas Operativos .

La evolución en sus diseños ó en su arquitectura de los computadores, que han significado un aumento en su velocidad de procesamiento así como aumentando su capacidad de almacenamiento, además del surgimiento y mejoras de los Sistemas Operativos, han influido grandemente en la generación de nuevas versiones de los lenguajes de programación así como el surgimiento de nuevos lenguajes.

Necesidad de nuevas aplicaciones.

Debido a los bajos costos de productos computacionales se ha incrementado la necesidad de nuevas aplicaciones en poco tiempo. Los requisitos ó especificaciones de las nuevas necesidades han sido tomados en cuenta para el diseño de los nuevos lenguajes de programación, y revisión de los ya existentes.

Metodologías de programación.

Las consideraciones respecto a características de los problemas donde las metodologías actuales de programación no han cubierto, respecto a escribir programas complejos se han visto reflejadas en el diseño de los nuevos lenguajes de programación.

Métodos de implementación.

El desarrollo de mejores métodos para la implementación de lenguajes ha permitido el diseño de mejoras al proceso de la implementación. Esto ha reducido notablemente los costos.

Estudios teóricos.

La investigación en áreas de la computación ha generado mejoras en el diseño de nuevos métodos de programación, asimismo de una visión clara de las debilidades de los métodos actuales. Esto ha abierto nuevos campos de trabajo en áreas de laboratorio probando aplicaciones de las nuevas técnicas de desarrollo, lo cual ha provocado la aparición de nuevos lenguajes de programación con nuevas metodologías de programación.

Estandarización.

Este factor de procurar estandarizar métodos mediante las consideraciones de las organizaciones internacionales ha influido grandemente, tratando que la transportabilidad de los programas entre diferentes computadores sea una realidad, ha sido un aspecto importante en la evolución de los lenguajes de programación.

A ciencia cierta existe un gran numero de lenguajes de programación para las computadoras, y algunos de los conocedores los agrupan bajo características de funcionamiento, aunque algunos lenguajes no encajan directamente en algún grupo en particular. Otros de los conocedores los clasifican por generaciones en similitud de la clasificación del hardware.

Clasificación de los lenguajes por sus características

- **Imperativos** (procedimentales)
  (Fortran,Cobol,Algol60,Algol68,Pascal,C,Modula,Ada,CLU)
- **Declarativos**
  - Funcionales(Lisp,APL)
  - Lógicos(Prolog)
- **Imperativos/Declarativos**
  (Lenguajes Orientados a Objetos y Lenguajes de 4ª Generación)

Los lenguajes imperativos son aquellos que usan principalmente la instrucción de asignación como el constructor básico y lo complementan las estructuras de control secuencial, estructuras alternativas e iterativas, su diseño se basa en la construcción de algoritmos de solución que son "seguidos" como una receta.

Una crítica generalizada a estos lenguajes es que idealmente el programador debe concentrarse en describir aquello que desea controlar en el programa sin preocuparse en como administrar el proceso de ejecución del programa, sobre todo con respecto al control de la utilización de la memoria.

Los lenguajes declarativos también son considerados declarativos/descriptivos e incluyen significados alternativos que describen valores de los datos y los cálculos.
Los lenguajes declarativos tienen tres características las cuales son: **Expresivos** por la manara que describen los problemas, situaciones, métodos ó soluciones. **Fiables** ya que tratan de proteger al programador tanto como sea posible en cuanto a cometer errores. **Elegantes** ya que son dirigidos al uso de un soporte matemático en el desarrollo de las aplicaciones.

Los lenguajes declarativos se clasifican en dos clases: Funcionales y lógicos.
Lenguajes Funcionales. Estos lenguajes son aquellos en que todas las aplicaciones son funciones en el sentido matemático, ó sea que no hay instrucciones, de forma que un programa funcional es una función que se define por descomposición de funciones mas simples. Estos lenguajes no usan la instrucción de asignación, la inexistencia de esta instrucción permite que el programador no se ocupe en el control de la memoria que está usando, esto le permite al programador concentrar sus esfuerzos en la descripción del problema.

Mientras que para los lenguajes orientados a instrucciones la forma de construir los programas es en base a organizar la ejecución en forma secuencial, así como de la de ejecución repetida de instrucciones(ciclos), todo ello a través del uso de estructuras de control adecuadas, en vez de ello la programación funcional hace combinación de funciones para generar otras mas potentes.

Otro punto mas respecto a los lenguajes funcionales es que tratan las funciones de la misma que cualquier otro valor, es decir pueden ser pasadas como parámetros, pueden ser el valor de una expresión ó pueden ser almacenadas en una estructura de datos.

Algunas veces el uso del término "funcional" es aplicado solamente para definir aquellos lenguajes que no usan en absoluto las características procedimentales, a éstos lenguajes se les considera "Lenguajes Funcionales puros" tales como el lenguaje Standard, el lenguaje ML, el lenguaje APL. Pero existen lenguajes funcionales que "si" usan la instrucción de asignación tales como el lenguaje LISP.

Lenguajes lógicos. La idea básica de la programación lógica se traduce a la expresión de la frase: "Algoritmos = Lógica + Control", esta frase muestra la forma de operar con estos lenguajes, y es opuesta a la forma de operar en los lenguajes tradicionales. El lenguaje lógico mas conocido es **Prolog**. Los lenguajes lógicos tratan con relaciones (predicados) entre objetos(datos), en lugar de hacerlo con funciones. Este principio se basa en la premisa que programar con relaciones permite mas flexibilidad en los programas que programar con funciones, ya que las relaciones tratan con argumentos y otorgan resultados uniformemente. Informalmente el uso de las relaciones permite ignorar direcciones de tal forma que no existe distinción entre los datos que se calculan y a partir de cuales.

Las relaciones se especifican con **"reglas"** y con **"hechos"**, **un hecho** sería una "frase" como : "Rosa es mujer" en Prolog sería mujer(rosa)
"Los Padres de Rosa son Carlos y Pilar" en Prolog sería padres(Rosa, Carlos, Pilar)
una "regla" se usa para expresar frases condicionales como:
"Dos personas son hermanas si son mujeres y tienen los mismos padres"
en Prolog sería : hermanas(x,y):- mujer(x), mujer(y), padres(x,p.m), padres(y,p,m).
Por tanto las reglas no sirven para deducir relaciones entre objetos que no se han definido explícitamente mediante hechos.

Imperativos / declarativos. Bajo esta clase se considera a los lenguajes orientados a objetos y también a los considerados lenguajes de cuarta generación. **Los lenguajes orientados a objetos** proponen una nueva metodología de programación caracterizado por una nueva concepción de los problemas a resolver mediante programas de

computadoras, esta nueva manera de programar se basa en tres conceptos nuevos: "clase", "objeto" y "herencia".

Una clase representa a un tipo de datos y es considerada una "clase de objetos", los objetos son entes dinámicos ya que pueden ser creados y borrados al momento de la ejecución del programa, los objetos de una misma clase mantienen características y procesos similares.

La herencia permite definir nuevas "clases de objetos" a partir de una clase previamente definida, la nueva clase heredará las características y propiedades de la clase previa.

Además de los conceptos mencionados(objetos,clases,herencia), se utilizan otros dos conceptos que son: "mensaje" y "método". El "mensaje" es usado para comunicar a un objeto la tarea que se espera que realice, es parecido a un procedimiento donde se especifica el objeto a quien va dirigido, el nombre de la operación a realizar, y los parámetros necesarios para realizarla.

El lenguaje orientado a objetos mas conocido es el Smalltalk el cual fue dado a conocer en 1976, a éste se ha añadido el lenguaje C++ a partir de 1986. Debido a que C++ es una extensión del lenguaje C los mensajes y métodos son directamente los usados para operación de funciones usuales en C, éste lenguaje paulatinamente ha incrementado su popularidad ya que su metodología es adecuada para tratar problemas de gran complejidad, así como el uso de la nueva tecnología emergente.

Los **lenguajes de cuarta generación** son aquellos que pretenden superar los problemas surgidos en los lenguajes de tercera generación tales como:

- Reducir el tiempo de construcción de las aplicaciones.
- Minimizar los problemas en la depuración de aplicaciones.
- Facilitar a los usuarios finales la forma de resolver sus propios problemas, usando la misma herramienta de programación.

- Generación automática de código ejecutable sin errores, a partir de definir los requerimientos deseados, y dándolos mediante expresiones de alto nivel.
- Reducir los costos de mantenimiento haciendo aplicaciones fáciles de hacer cambios.

Generalmente los lenguajes de cuarta generación están orientados al tratamiento de grandes cantidades de información, mediante el uso de grandes Bases de datos.

Algunos de los lenguajes de esta clase son considerados como de propósito general tales como: IDEAL, MANTIS, NATURAL, otros lenguajes de esta misma clase pero considerados como especialmente al tratamiento de datos son: SQL, QBE ó ADRS.

Otro punto de vista en la clasificación es considerando que los lenguajes de programación han sufrido transformaciones que se engloban bajo el concepto de generaciones.

- Lenguajes de primera generación.
- Lenguajes de segunda generación
- Lenguajes de tercera generación
- Lenguajes de cuarta generación
- Lenguajes de quinta generación

Las cuales se comentarán en el siguiente capitulo.

3

Características de las generaciones de los lenguajes de programación

3.1 Generaciones de los lenguajes de programación.

De todos los inmersos en el ámbito de los Sistemas Computacionales(también llamado ámbito de la Informática, y de la Computación), es sabido que una computadora digital se concibe en forma general bajo dos conceptos a saber: Hardware y Software.

El hardware se refiere a todos los componentes físicos que forman la máquina que llamamos computadora(Gabinete, Monitor, Tarjetas de circuitos impresos, cables, etc..), en suma son "todas aquellas partes que podemos ver y tocar".

El software se refiere a los elementos intangibles ó sea aquellas partes que no podemos tocar(los programas), pero que al ser "integradas" al hardware se logra que la maquinaria inerte "adquiera vida artificial" y pueda realizar funciones de procesamiento de datos. Metafóricamente el software para el hardware es como el combustible para el automóvil, un elemento que no es parte integral de él, pero que le es indispensable para su funcionamiento.

El software ha sido un pilar para la utilización efectiva del hardware, y los programadores son los artífices "creadores", sus herramientas principales son :

- Los lenguajes de programación(creados por ellos mismos), donde se definen los métodos y técnicas de la programación.
- Su propia imaginación.

mediante los cuales logran el desarrollo de aplicaciones que sean soluciones a los diferentes tipos de problemas, a través de darle "ordenes" a la computadora para que ejecute las tareas deseadas.

Al igual que el hardware el software ha evolucionado a través del tiempo, y a través de él se ha logrado que el ser humano "se comunique" con la computadora y le pueda indicar que realice actividades ó tareas definiéndole claramente qué hacer, como hacerlo y cuando hacerlo.

Por medio del software(Lenguajes de programación) se hace uso de las características del hardware y mediante el acceso a la memoria de la máquina se deja (almacena) en ella los datos con los que trabajará y las instrucciones que le indiquen el procedimiento a seguir para la realización de la tarea deseada.

3.1.1 Lenguajes de primera generación.

En esta generación se considera a los **lenguajes de máquina y los lenguajes ensambladores**. Cada arquitectura de computadora tiene un solo lenguaje de máquina que "entiende", por lo que existen lenguajes de máquina tantos como arquitecturas de computadoras existan. **Los lenguajes de máquina** tienen un conjunto reducido de símbolos numéricos mediante los cuales se indican las acciones a realizar y los datos que intervienen en las operaciones. Ya que los símbolos que utiliza son números sus instrucciones son una secuencia de números, que para otras personas fuera del programador(de lenguaje de máquina) son "algo in entendible". El programar en este lenguaje se tienen probabilidades de un alto grado de error.

Ejemplo.

41 3 0C1A4

3A 2 0C1A8

1A 3 0C1A0

50 3 0C1A4

esta secuencia representa la instrucción:

$$z = w + x + y$$

un programa en lenguaje de máquina, es el medio mediante el cual el programador "directamente" le indica a la computadora las acciones de la tarea a realizar.

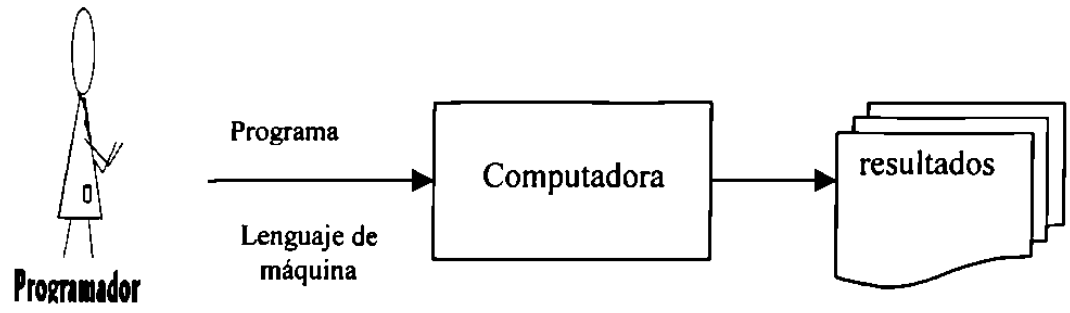


Fig. 2 Esquema del Proceso de la programación en lenguaje de máquina

El **"lenguaje de máquina"** es también llamado **"lenguaje binario"**, ya que sus instrucciones son dadas directamente mediante mecanismos de tipo switch(apagado=0, encendido=1), fue el primer lenguaje de programación ya que nació junto con la primera computadora, y de algunas de sus características se pueden mencionar:

- Dependiente de la máquina a la que se dirige el programa.
- Los programas No tienen transportabilidad.
- La programación es sumamente difícil.
- El período de enseñanza-aprendizaje es muy largo

La programación en lenguaje de máquina por lo difícil de definir sus instrucciones es sumamente tediosa, ya que el programador debe "bajarse" al nivel de entendimiento de la máquina para indicarle paso a paso y en forma numérica las acciones a realizar. Esta tarea es muy propensa a errores y la productividad del programador era muy limitada, pero también hay que reconocer que en ese tiempo era la única alternativa existente.

Debido a la problemática del lenguaje de máquina se diseñó un nuevo lenguaje que permitiera reducirla y surge el **lenguaje ensamblador,** el cual es la representación simbólica del lenguaje de máquina, y significó un avance al intentar reducir las dificultades de la programación, ya que contiene un conjunto de símbolos alfabéticos como equivalentes a los símbolos numéricos de los lenguajes de máquina, solo en el direccionamiento de los operandos se usan símbolos numéricos. Esto redujo la propensión a errores de parte del programador, sin embargo la programación en lenguaje ensamblador requiere de conocer perfectamente la arquitectura del computador para el cual se construye el programa. Cada lenguaje de máquina tiene su lenguaje ensamblador asociado.

Tanto los lenguajes de máquina como los lenguajes ensambladores son considerados "dependientes de la máquina".

El **lenguaje ensamblador** requiere de una interfaz de conversión al lenguaje de máquina para que pueda ser ejecutado, esa interfaz es llamada precisamente "programa ensamblador". El programador codifica el programa(las instrucciones) de acuerdo a los símbolos del lenguaje ensamblador y esa codificación considerada el "programa fuente" y generalmente es almacenada como un archivo. Con el "programa fuente" se realiza el proceso de "traducción" al lenguaje de máquina, y el "traductor" es precisamente el lenguaje "ensamblador" que genera un archivo en lenguaje de máquina para ser ejecutado en la computadora.

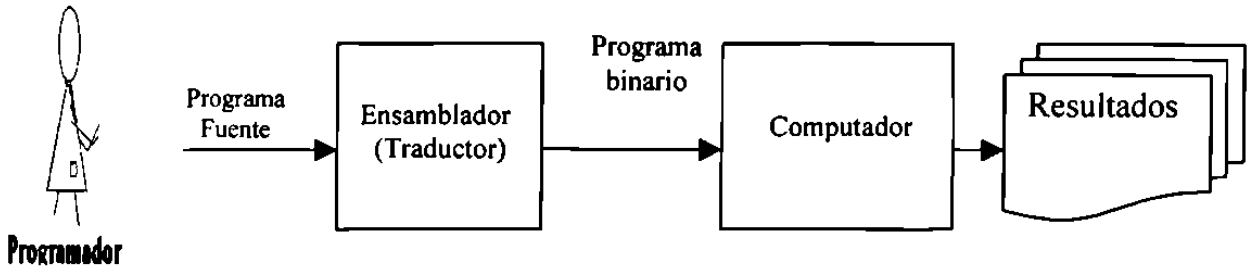


Fig. 3 Esquema del Proceso de la Programación en lenguaje ensamblador

El lenguaje ensamblador redujo los errores triviales, como puede ser el numero que corresponde a una operación, los cuales son difíciles de detectar pero fáciles de cometer, sin embargo aún en éste lenguaje es fácil para el programador perderse y cometer errores de lógica, ya que debe de "pensar" ó "bajarse" al nivel de la forma en como trabaja el CPU, y entender todo lo que sucede dentro del mismo CPU.

Sus características son parecidas al lenguaje de máquina :

- Dependiente de la máquina a la que se dirige el programa.
- Los programas No tienen transportabilidad.
- La programación sigue siendo tediosa y difícil.
- El período de enseñanza-aprendizaje se reduce comparado con el lenguaje de máquina.

Un ejemplo de los símbolos que se usan en instrucciones del lenguaje ensamblador son los siguientes, los cuales son para el microprocesador 80386/80286:

Símbolo	Significado	Ejemplo
ADD	Suma	ADD AX,AX
MOV	Copia una fuente a un destino	MOVE AX,TABLA_1
MUL	Realiza la multiplicación	MULT VAL_X
NOP	NO OPERACIÓN	NOP
PUSH	Carga palabra en pila	PUSH EBX
INC	Incrementa en 1	INC AX
IN	Entra Byte ó palabra	IN AX W_P_ADR
JA	Bifurca si anterior	JA INST_LABEL
JMP	Bifurcación incondicional	JMP FAR_LABEL
LAR	Carga byte acceso correcto, Selector	LAR ARB,SELECTR
LEA	Carga dirección efectiva de Desplazamiento	LEA AX,[BP][DI]
LOCK	Activa la señal BUS LOCK	LOCK MEM_WORD,AX
LOOP	Control de lazo con Contador	LOOP AGAIN
FIDIV	División entera	FIDIV WORD_INT
FINCSTP	Incrementa puntero de pila	FINCSTP

Respecto a la clase de lenguajes procedurales se considera a los lenguajes Fortran I y Algol58 como los lenguajes procedurales de primera generación, los cuales fueron creados entre 1954 y 1958, los cuales se basaban en expresiones matemáticas.

3.1.2 Lenguajes de segunda generación.

En las décadas de 1950 y 1960 inició el desarrollo de algoritmos de alto nivel, y científicos de otras ramas del conocimiento tales como la física y la química desearon ingresar al uso de la herramienta de la computadora para aplicaciones según su especialidad, por lo que se generó para ellos un nuevo "traductor" al cual se le dio el nombre de "compilador", con él nacieron los lenguajes de alto nivel y el primer compilador fue para el lenguaje FORTRAN. Esto significó un nuevo esfuerzo en la búsqueda de una mayor abstracción del lenguaje de máquina, facilitando la tarea del programador en la construcción de sus programas y aumentando su productividad.

La aparición de los lenguajes procedurales FORTRAN II, COBOL y ALGOL60(1959-1961), marcaron el surgimiento de la segunda generación de lenguajes de programación de computadoras, estos lenguajes realizaban el proceso de "traducción", mediante el "traductor" llamado " Compilador".

El Compilador además del proceso de "traducción" incluye otro proceso donde utiliza una nueva herramienta llamada "Linker", que utiliza un nuevo concepto llamado "funciones de biblioteca", los cuales son subprogramas que realizan tareas específicas tales como :

- Calcular una raíz cuadrada
- Calcular el valor absoluto de un valor numérico
- Redondeo de un valor numérico

etc..., y estos subprogramas son incluidos en los archivos del compilador, almacenados en una biblioteca ó librería de donde son extraídos al momento del proceso de "compilación" e insertado su código de máquina en los lugares donde son enunciadas en las instrucciones del programa fuente. La utilización de "Compiladores" es un esfuerzo por lograr que los programas de aplicación sean independientes de la máquina donde son realizados y puedan ser ejecutados en cualquier computadora digital.

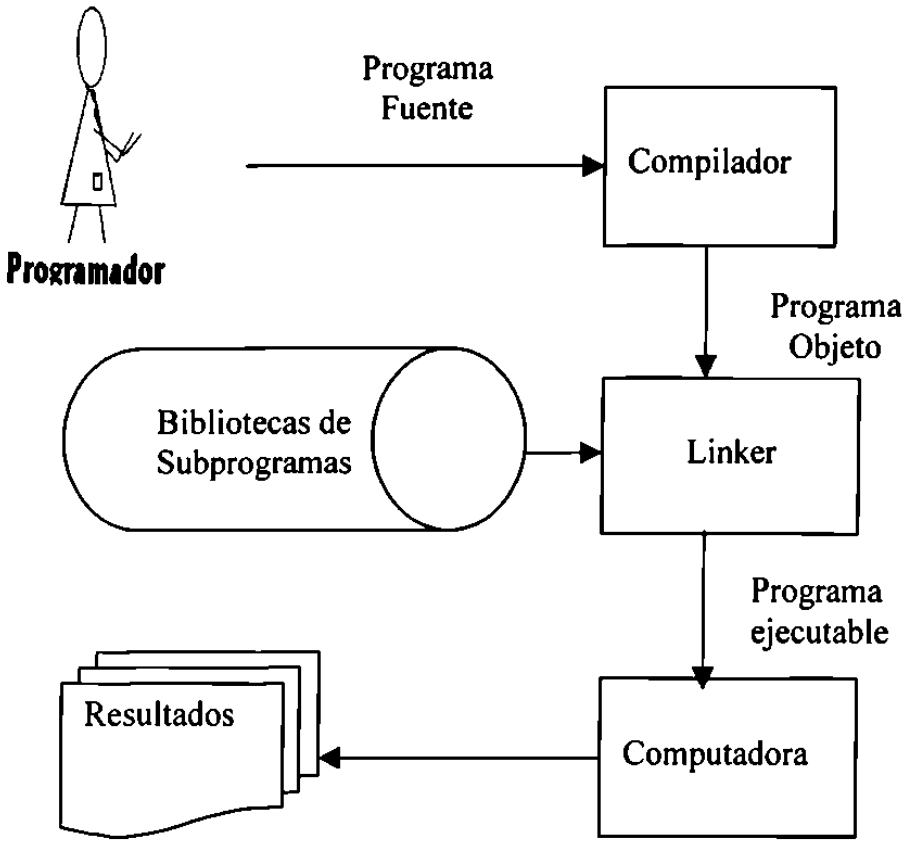


Fig. 4 Esquema del Proceso general de compilación y ejecución de un programa

En este punto es importante señalar que los programadores no seguían ninguna técnica estandarizada en el desarrollo de programas, cada programador "creaba" sus aplicaciones bajo su muy particular punto de vista(lógica), y no se acostumbraba generar una documentación de la aplicaciónes, por lo que el desarrollo, depuración y mantenimiento eran sumamente lentos y caros. Un problema fuerte para las instituciones publicas y privadas era la rotación de especialistas computacionales, ya que esto dificultaba aún mas los mantenimientos de las aplicaciones.

3.1.3 Lenguajes de tercera generación.

La tercera generación en los lenguajes de programación emerge con la aparición de lenguajes como: ALGOL68, PL/I, y PASCAL (1962 a 1970), así como una nueva versión del lenguaje Cobol estructurado, los cuales continúan con el uso de compiladores para la generación de los programas en lenguaje de máquina, con los lenguajes de ésta generación surgen las técnicas de "Programación Modular" y la "Programación Estructurada", con lo que se intenta crear escuela de programadores con alta eficiencia en el desarrollo de aplicaciones, surge también el concepto de "grupos de desarrollo", donde con las nuevas técnicas se hace posible que un programa pueda ser desarrollado por varios programadores, a través del uso de descomposición modular del problema general.

La técnica de programación estructurada de origen fue diseñada para facilitar el desarrollo de aplicaciones grandes y complejas. Al final de la década de 1970 emergieron numerosos lenguajes de diferentes tipos, en ese tiempo aproximadamente el 85% de los programadores desarrollaban sus aplicaciones en lenguaje Cobol.

En esta generación surge una gran cantidad de lenguajes y algunos los clasifican como **procedimentales** y **declarativos** así como al tipo de aplicaciones a que son dirigidas:

- Científicos(Fortran y APL).
- De gestión de datos(Cobol y RPG).
- De Aplicaciones de propósito general(Basic, Pascal,PL/I).
- Educativos(Logo,Pilot).
- Programación de Sistemas(C,ADA).
- De Inteligencia Artificial(Lisp,Prolog).

También surgen las teorías de **Lenguajes Orientados a Objetos**: (Smalltalk).

La **"Programación Modular"** y la **"Programación Estructurada"** son técnicas muy usadas en la actualidad, pero la aparición de nueva tecnología ha aumentado la complejidad de las aplicaciones por lo que ésta ha rebasado a las técnicas mencionadas.

Los lenguajes de tercera generación (ó lenguajes de "alto nivel") son lenguajes que permiten expresar los algoritmos en forma fácil, legible y comprensible para otros programadores, además el uso de compiladores le otorga las siguientes características:

- Independiente de la máquina de la que se dirige el programa.
- Los programas Si tienen transportabilidad.
- La programación se facilita siguiendo una metodología de desarrollo estándar.
- Se reduce considerablemente el período de enseñanza-aprendizaje

Algunos lenguajes de ésta generación usan otro tipo de "traductor" llamado "interprete", el cual tiene como característica que no genera código ejecutable sino que genera un código intermedio llamado pseudocódigo, que se ejecuta al momento de "interpretar" la acción a realizar, pero que al terminar la ejecución del programa el pseudocódigo se pierde. Esta forma de operar obliga a que en cada ejecución de un programa se realice la "traducción" directamente del programa fuente.

3.1.4 Lenguajes de cuarta generación.

Cuando los programadores empezaron a usar metodologías estandarizadas inició también la documentación realizando sus algoritmos bajo las técnicas de: diagramas de flujo, Pseudocódigo, Nassi-Schneiderman, y otras mas. En la década de 1980 cuando estuvieron disponibles las PC's y las interfaces gráficas algunas compañías usaron herramientas gráficas que les permitieran analizar y crear sus diagramas estructurados directamente en la computadora y guardadas en archivos.

Por el mismo tiempo otras personas crearon herramientas que generaban código directamente a partir de diagramas estructurados, ya que cada símbolo gráfico de una estructura gráfica "significaba" ciertas palabras que eran traducidas a una ó mas instrucciones de algún lenguaje procedural. Al conjuntar las herramientas de gráficas de diagramas estructurados, y la generación automática de código se creó una nueva herramienta llamada CASE(Computer-Aided Software Engineering). Algunos proveedores de Software CASE promocionan este producto como una herramienta de desarrollo de cuarta generación.

Los lenguajes de esta generación intentan superar los problemas de los lenguajes de la tercera generación, haciendo manejable la complejidad de las aplicaciones usando la nueva tecnología computacional. Con los lenguajes de cuarta generación el programador hace uso de un conjunto de instrucciones secuenciales así como una gran variedad de mecanismos como formularios para la interacción en pantalla.

El uso de Bases de datos relacionales generó la posibilidad de almacenar una gran cantidad de datos que fueran independientes de cualquier aplicación, de forma que pudiesen generarse aplicaciones cortas y fáciles que accesaran y explotaran la información contenida en las Bases de datos. A los lenguajes que permiten realizar lo anterior son considerados los lenguajes de cuarta generación(4Gls).

Los lenguajes de cuarta generación (4Gl's) permiten acceder Bases de datos relacionales.

Algunos lenguajes de esta generación están basados en cuestionarios, otros son generadores de documentos ó de gráficos, con ellos el usuario estipula lo que desea realizar y el lenguaje se encarga él solo de definir "cómo" hacerlo.

Algunos de los lenguajes de la cuarta generación son: MANTIS, IDEAL, NATURAL, éstos lenguajes son de propósito general. SQL, y QBE son lenguajes dirigidos a consultas de grandes Bases de datos.

3.1.5 Lenguajes de quinta generación.

Los lenguajes de esta generación permiten al programador el ingreso de "hechos" y consultas, sin especificar ningún algoritmo para obtener la respuesta deseada.

Estos lenguajes permiten el diseño de programas que emulan un comportamiento inteligente, sus aplicaciones son consideradas campos de la Inteligencia Artificial(IA).

LISP y PROLOG son los lenguajes prototipo de ésta generación de lenguajes ya que facilitan aplicaciones como : juegos de ajedrez, aplicaciones para diagnósticos médicos, diagnósticos mecánicos etc.

Algunas de las aplicaciones ó áreas de la Inteligencia Artificial son:

- Sistemas Expertos
- Robótica
- Procesamiento de lenguajes naturales
- Visión Artificial
- Modelos del conocimiento

3.2 Panorama histórico de los lenguajes de programación.

Una visión retrospectiva de los lenguajes de programación nos permite visualizar su evolución histórica y de este modo apreciar sus características diferentes en el tiempo, los lenguajes mas modernos actualmente nos aconsejan no usar en lo posible la instrucción: goto, para acciones de bifurcación. Esto es correcto de acuerdo a la metodología de la programación estructurada y a la ingeniería de software.

A este respecto conviene recordar que hubo un tiempo en que la instrucción goto en combinación con el if eran la única herramienta disponible para el programador, ya que no existían instrucciones como el for,while ó el if-then-else.

Algo importante que nos permite la historia es ver la evolución de familias de lenguajes, así como la influencia que ejercen las arquitecturas de las computadoras en el diseño de los lenguajes de programación, y evitar futuros defectos de diseño aprendiendo de los diseños anteriores.

En seguida se muestra una gráfica donde muestran algunos de los lenguajes de programación existentes hasta la década de 1980, hubo algunos lenguajes que no se hicieron populares y que tenían nombres muy diversos tales como: MAD, AMBIT, BASEBALL. Los lenguajes que se mencionan se eligieron por su gran aceptación y uso entre los programadores así como por sus características.

Las líneas continuas que hacen relaciones entre los lenguajes indican ascendencia directa, mientras que las líneas discontinuas indican solo influencia. Los lenguajes prefijados con las letras ANS indican que este lenguaje ha sido adoptado por la American National Standars Institute(ANSI), como un estándar nacional., reforzando de esta forma la "Transportabilidad" de programas de una máquina a otra, ya que todos los proveedores de computadoras deberán implementar este lenguaje en sus productos.

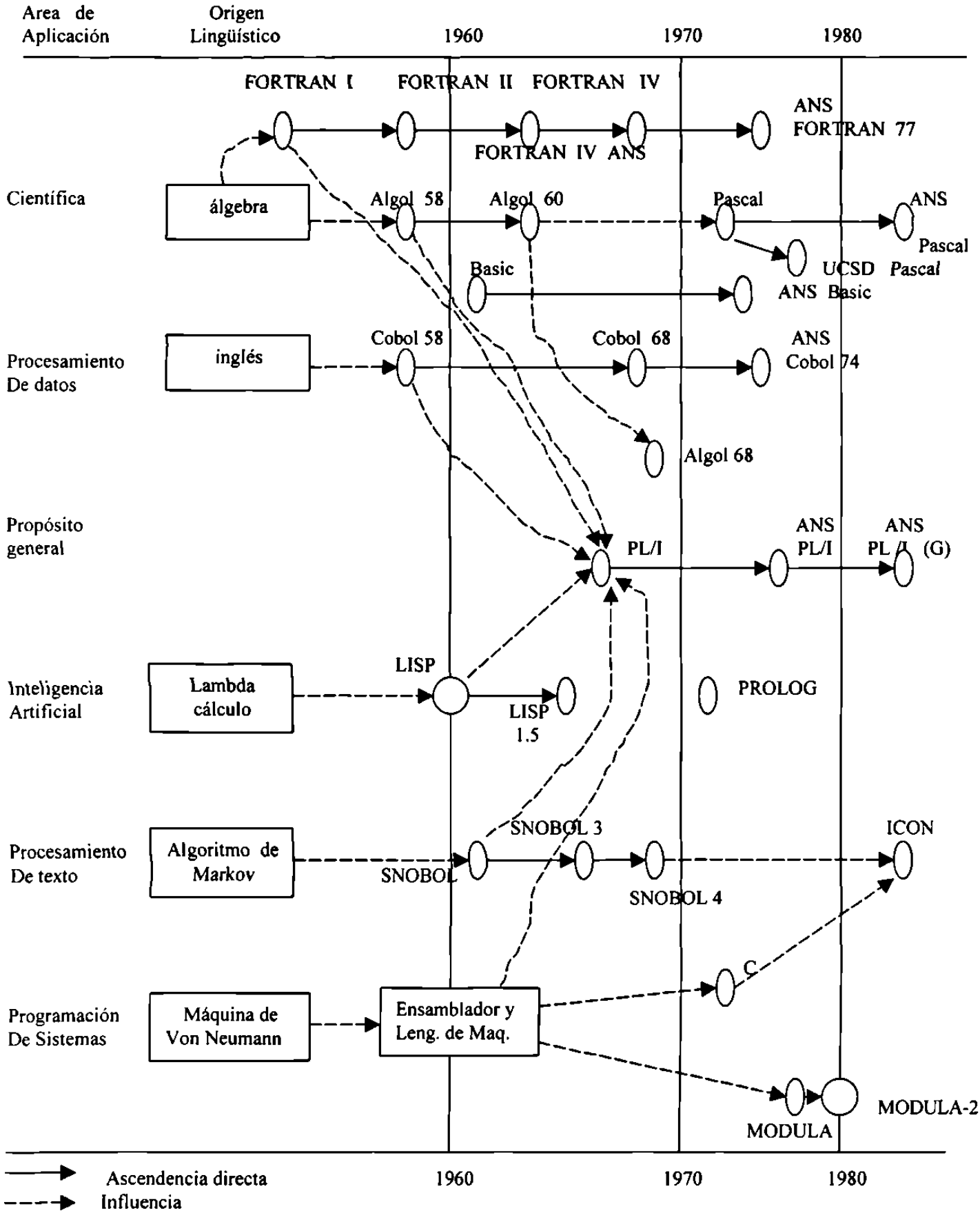


Fig. 5 Perspectiva histórica de algunos lenguajes de programación

Simula-67 fue el lenguaje que introdujo los conceptos de "clase" y "herencia" los cuales fueron adoptados por el lenguaje Smaltalk, éste lenguaje aparece comercialmente en 1976 y tuvo una gran aceptación con su versión Smaltalk-80, pero su versión mas reciente la Smaltalk/V de la empresa Digital ha tenido una buena aceptación ya que opera bajo Windows, con un ambiente de desarrollo integrado y operado mediante menús.

La mayoría de los lenguajes orientados a objetos que se han desarrollado han sido en base a lenguajes ya tradicionales como Basic, C++, Objetive-C Modula-2, Object Pascal y recientemente Object Cobol.

De los lenguajes orientados a objetos mas recientes y mas populares por su versatilidad y su gran poder de cómputo es el lenguaje Java de Sun Microsystems.

Existen lenguajes orientados a objetos que no se han popularizado tales como el lenguaje Eiffel que soporta todas las propiedades y fundamentos del ambiente de objetos, y solo es usado en ambientes universitarios y de investigación, aunque su versión mas reciente ya opera en ambiente Windows y se espera que aumente su difusión. Ada es otro lenguaje que con su versión mas reciente Ada-95 ya soporta los conceptos de herencia y polimorfismo.

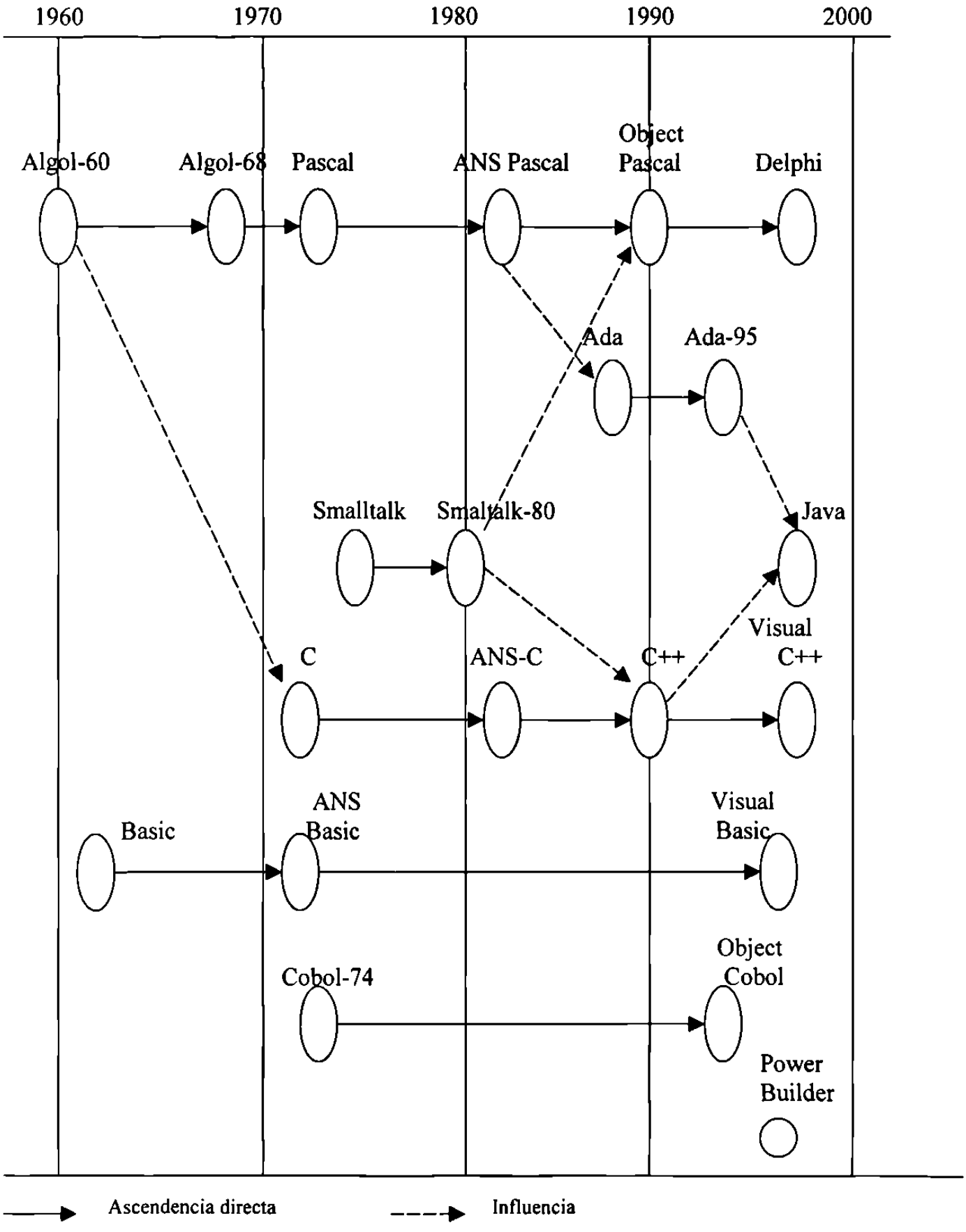


Fig. 6 Perspectiva histórica de los Lenguajes Orientados a Objetos

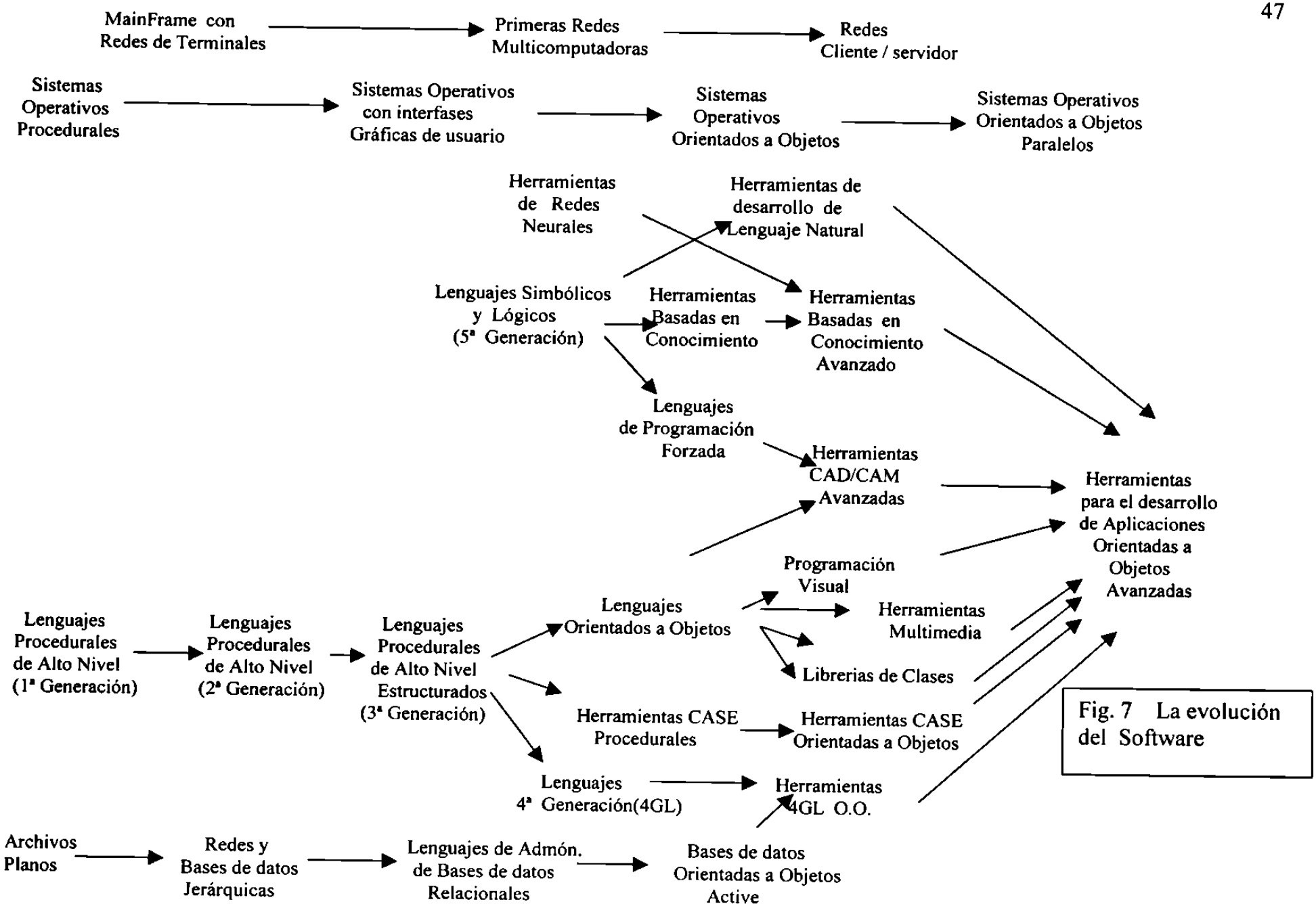


Fig. 7 La evolución del Software

4

IMPORTANCIA DE LA ENSEÑANZA DE TÉCNICAS ALGORITMICAS

Uno de los rasgos del perfil del profesionista del área de Sistemas Computacionales es la capacidad de razonamiento analítico de los problemas, ya que ésta será la forma en que abordará los diversos problemas que se le plantearán en el ámbito de sus actividades. El término analítico implica la descomposición de un todo haciendo distinción de los elementos constitutivos(ó partes), haciendo de esta actividad un método de solución de problemas. Los alumnos de Ingeniería en Sistemas Computacionales ejercitan esta cualidad mediante el aprendizaje del desarrollo de programas de computadoras para la solución de diversos tipos de problemas. En el aprendizaje de la programación de computadoras los alumnos se enfrentan a dos problemas:

- Aprender a analizar el problema a resolver y formular el procedimiento de solución, éste debe quedar definido dentro del "Marco Computacional" ó sea en acciones que sean fáciles de de traducir a una computadora.
- Debe aprender a comunicar a la computadora el procedimiento de solución que ha desarrollado, mediante "órdenes" que entienda y ejecute la computadora.

El primer problema requiere de entusiasmo, disciplina, desarrollo de una lógica para la aplicación del método analítico, y temple para no darse por vencido ante los intentos fallidos, el segundo problema requiere del conocimiento de los "comandos" de operación de un lenguaje computacional específico.

4.1 Técnicas Algorítmicas.

La solución de problemas mediante el uso de computadores es un rasgo que caracteriza a los profesionistas de la carrera de Ingeniería en Sistemas Computacionales, esto lo realizan a través de generar aplicaciones ó programas orientados a problemas específicos.

Para la generación de aplicaciones ó programas se sigue un procedimiento de Programación que incluye las siguientes fases:

- Análisis del Problema a resolver
- Diseño del Algoritmo de solución
- Codificación del Algoritmo en un Lenguaje Computacional específico
- Ejecución y validación de resultados
- Liberación ó puesta en producción
- Mantenimiento

La observancia estricta del orden descrito de las fases es requisito fundamental para que las aplicaciones generadas logren los resultados deseados.

Las fases 1 y 2 son para la Programación de computadoras lo mismo que los cimientos para un edificio: la base que sostiene toda la estructura, si la base es sólida soportará todas las pruebas a que sea sometida, en caso contrario todo lo que se construya sobre ella no podrá sostenerse.

La fase del Análisis del problema tiene como objetivo discernir alternativas ó procedimientos de solución al problema planteado, y seleccionar la alternativa mas factible.

La fase de Diseño del Algoritmo tiene como objetivo representar en forma gráfica las acciones a realizar para lograr la solución deseada.

Definición del término Algoritmo.

Un Algoritmo es una secuencia de instrucciones a seguir para resolver un problema específico.

Para los aprendices de la programación de computadoras(alumnos), es obligatorio "aprender" y "dominar" primeramente las fases:

- Análisis del problema a resolver
- Diseño del Algoritmo de solución

del procedimiento para la programación de computadoras, estas fases son independientes de lenguaje computacional.

Se disponen de tres Técnicas Algorítmicas para la representación de los algoritmos a saber:

- Diagramas de Flujo
- Diagramas Nassi-Schneiderman
- Pseudocódigo

Las Técnicas Algorítmicas son herramientas que tienen por objetivo facilitar el diseño de soluciones de problemas orientadas al uso de la computadora.

4.2 Antecedentes Plan reticular ISIC-1993-296

El plan reticular ISIC-1993-296 de Ingeniería en Sistemas Computacionales tiene definida la enseñanza del Procedimiento de la Programación en forma secuenciada, ya que define primeramente el aprendizaje del método analítico en la solución de problemas en el primer periodo escolar a cursar con la materia "Diseño estructurado de Algoritmos", y posteriormente en el siguiente periodo escolar el aprendizaje de un lenguaje de programación para el desarrollo de soluciones mediante el uso de la computadora.

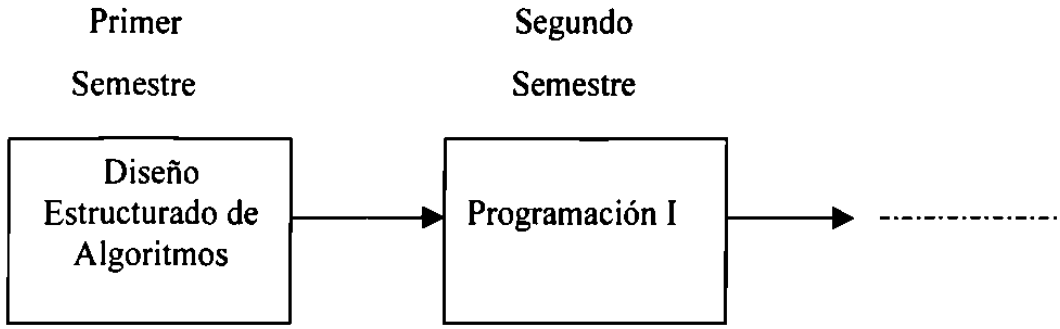


Fig. 8 Secuencia de la enseñanza de la programación a alumnos de I.S.C. en los 2 Primeros semestres.

Nota. La línea continua en flecha significa que la materia precedente es requisito aprobar para poder cursar la siguiente materia en secuencia.

Este diseño ha sido definido tomando como base que en los planteles escolares de niveles precedentes al universitario, no han ejercitado a los alumnos en el método analítico para la solución de problemas mediante el uso de la computadora.

La materia de "Diseño Estructurado de Algoritmos" tiene el objetivo de lograr que el alumno sea capaz de plantear metodológicamente la solución a problemas susceptibles de ser computarizados, mediante la aplicación del Análisis, asi como la enseñanza de diversas técnicas para representar el diseño de la solución .

La materia en si no es una materia fácil, ya que en ella los alumnos se enfrentan al problema de describir en forma lógica paso a paso las operaciones que se deben realizar para dar solución a un problema específico, sin hacer referencia a algún lenguaje computacional, sino solo en el planteamiento del problema y el diseño Algorítmico de la solución.

Esta actividad los alumnos no están acostumbrados a realizarla, y por lo tanto los ejercicios que se plantean "se les hacen muy complicados".

El tomar costumbre de realizar la actividad del análisis del problema como el inicio para obtener el diseño de la solución es un proceso lento, la parte práctica de este proceso es la representación gráfica del algoritmo de solución mediante alguna de las técnicas mas conocidas(Diagrama de flujo, Pseudocódigo, Diagramas N-S), asï de esta forma cada alumno desarrolla esta habilidad en forma individual, facilitando que cada uno de ellos puede desarrollar "su propia lógica" en el diseño y representación de la solución del problema planteado.

Sin embargo cuando por alguna situación los alumnos no le dedican el tiempo suficiente a ejercitar esta habilidad, y no se logra el objetivo del desarrollo de la "lógica" individual en el diseño de soluciones a los problemas, como resultado de esta problemática se tiene un alto nivel de reprobación en los alumnos que cursan por primera vez esta materia.

La materia "Programación I" tiene el objetivo de que el alumno continúe con el aprendizaje del las Fases del Procedimiento para la Programación, ejercitándolo en todas las fases del mismo, en esta materia se asume que el alumno "ya" aprendió a desarrollar algoritmos de solución a los problemas planteados, por lo que ahora aprenderá código y sintaxis de un lenguaje computacional.

La materia de "Diseño Estructurado de Algoritmos" es muy importante para que los alumnos logren un buen aprovechamiento del contenido de la siguiente materia en secuencia ó sea "Programación I".

Si la materia "Diseño Estructurado de Algoritmos" no se cursa adecuadamente podría ser causa de un alto índice de reprobación en "Programación I", y aún mas si no se incluyera en el plan reticular de la carrera.

Se podría hacer un comparativo de los alumnos de otras carreras de Ingenierías que también cursan la materia de Programación, similar a la materia de Programación I de la carrera de Ingeniería en Sistemas Computacionales.

4.3 Información Estadística de Alumnos de Primer Ingreso.

Para este trabajo de tesis se realizó una pequeña investigación sobre los 6 semestres inmediatos pasados (de feb-Jun'1998 a Ago-Dic'2000), sus resultados se muestran en la siguiente gráfica.

Materia: Diseño Estructurado de Algoritmos

	A-D '00	F-J '00	A-D '99	F-J '99	A-D '98	F-J '98
Alumnos Primer Ingreso	163	120	198	106	153	50
Alumnos Aprobados	93	37	98	59	64	18
Alumnos Reprobados	70	83	100	47	89	32

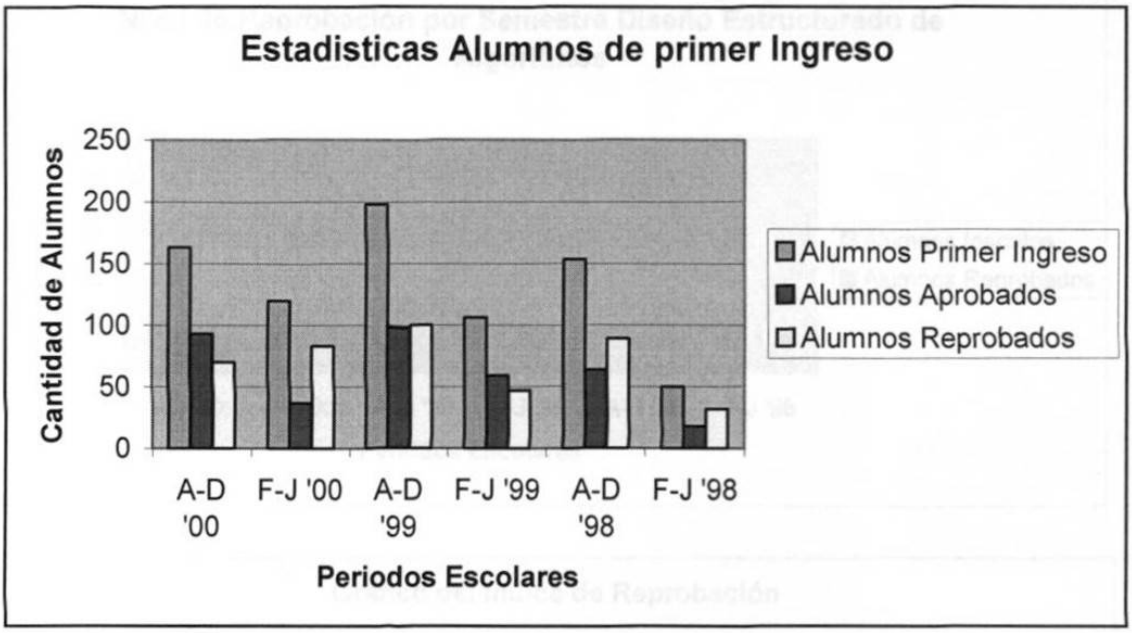


Fig. 9 Estadísticas del Nivel de reprobación en Diseño Estructurado de Algoritmos

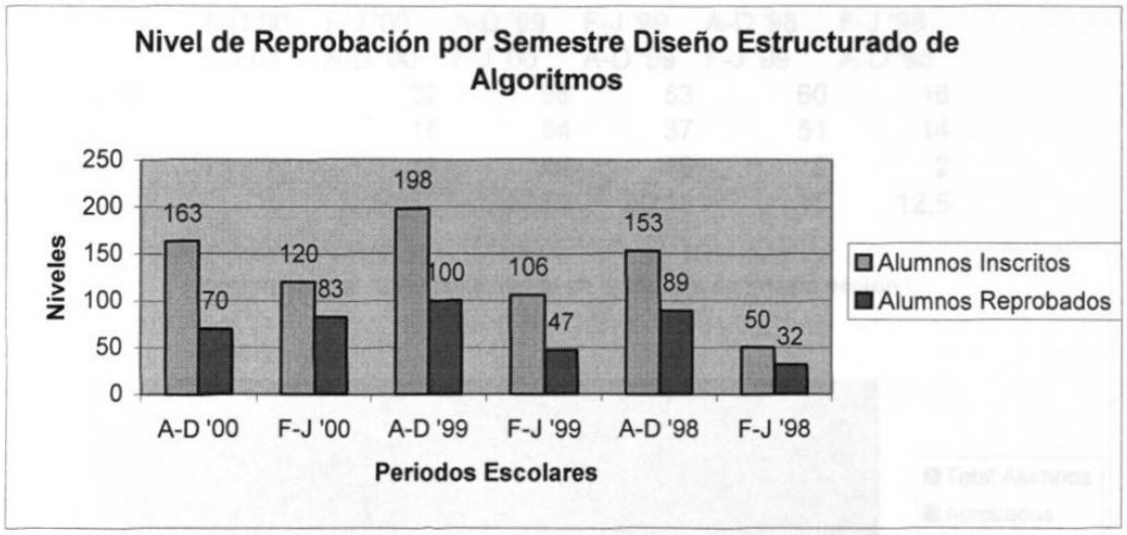
El significado de las claves son:

- A-D Agosto-Diciembre
- F-J Febrero-Junio
- '00 Año 2000
- '99 Año 1999
- '98 Año 1998

Como puede observarse en la gráfica el nivel de reprobación en la materia mencionada es muy alto, si calculamos el índice de reprobación que le corresponde a cada Periodo Escolar quedaría de la siguiente manera:

	A-D '00	F-J '00	A-D '99	F-J '99	A-D '98	F-J '98
Alumnos Inscritos	163	120	198	106	153	50
Alumnos Reprobados	70	83	100	47	89	32

	A-D '00	F-J '00	A-D '99	F-J '99	A-D '98	F-J '98
Indice de Reprobacion	42,9448	69,167	50,5051	44,34	58,1699	64



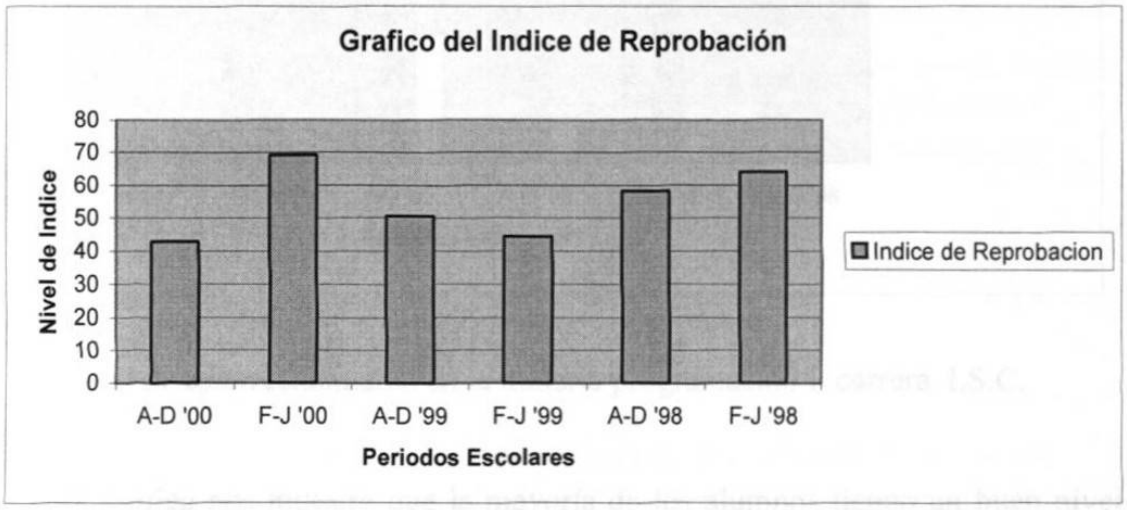


Fig. 10 Indice de reprobación en materia Diseño Estructurado de algoritmos

Se concluye que el índice de reprobación es alto en "Diseño Estructurado de Algoritmos" como primer materia relacionada con la profesión de Ingeniero en Sistemas Computacionales, sin embargo para los alumnos que logran aprobar esta materia los resultados del aprendizaje en la programación de computadoras se refleja en la siguiente materia en secuencia: "Programación I".

En la materia "Programación I" el alumno aprenderá código y sintaxis de un lenguaje computacional, y con ello ejercitará "**Todas**" las fases del Proceso de programación desarrollando programas para diversos tipos de problemas.

	A-D 00 F-J 01	F-J '00 A-D '00	A-D '99 F-J '00	F-J '99 A-D '99	A-D '98 F-J '99	F-J '98 A-D '98
Total Alumnos		32	98	53	60	16
Aprobados		15	54	37	51	14
Reprobados		17	44	16	9	2
Indice Reprob.		53,12	44,89	30,18	15	12,5

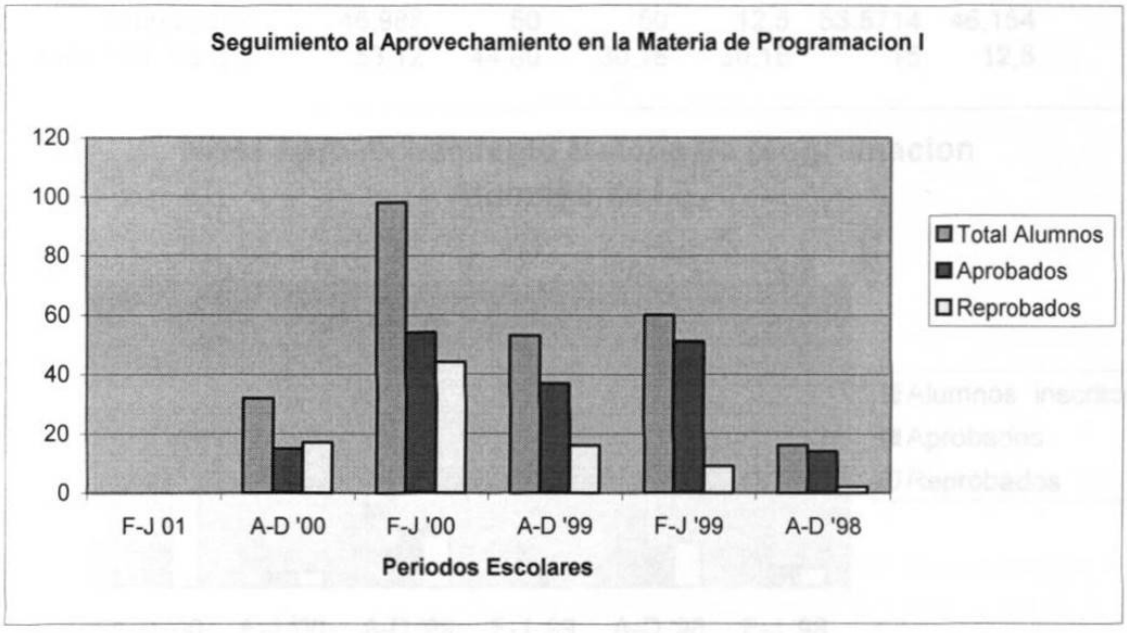


Fig. 11 Nivel de aprovechamiento en la materia programación I carrera I.S.C.

Esta gráfica nos muestra que la mayoría de los alumnos tienen un buen nivel de aprovechamiento en Programación I, y esto es debido a que al cursar esta materia ya traen un background importante en al Análisis de problemas y Diseño algorítmico de su solución.

4.4 Comparativo del Nivel de Aprovechamiento en Programación.

Otras Carreras Profesionales también incluyen en su retícula de estudios la materia de Programación, pero no incluyen el precedente de "Diseño de Algoritmos", sino que directamente tratan el aprendizaje de código y sintaxis de un lenguaje computacional, dando poco énfasis en el Análisis del problema y el diseño algorítmico de la solución.

Para este trabajo de tesis se hizo una pequeña investigación sobre el nivel de aprovechamiento de la materia de "Programación" en los alumnos de la carrera de Ingeniería Electrónica, y su comparativo con el nivel de aprovechamiento de su equivalente "Programación I" de la carrera de Ingeniería en Sistemas Computacionales.

	A-D '00	F-J '00	A-D '99	F-J '99	A-D '98	F-J '98
Alumnos inscritos	83	10	34	16	28	13
Aprobados	44	5	19	14	13	7
Reprobados	39	5	17	2	15	6
Indice de Reprobacion	46,988	50	50	12,5	53,5714	46,154
Indice Reprob. I.S.C.	53,12	44,89	30,18	30,18	15	12,5

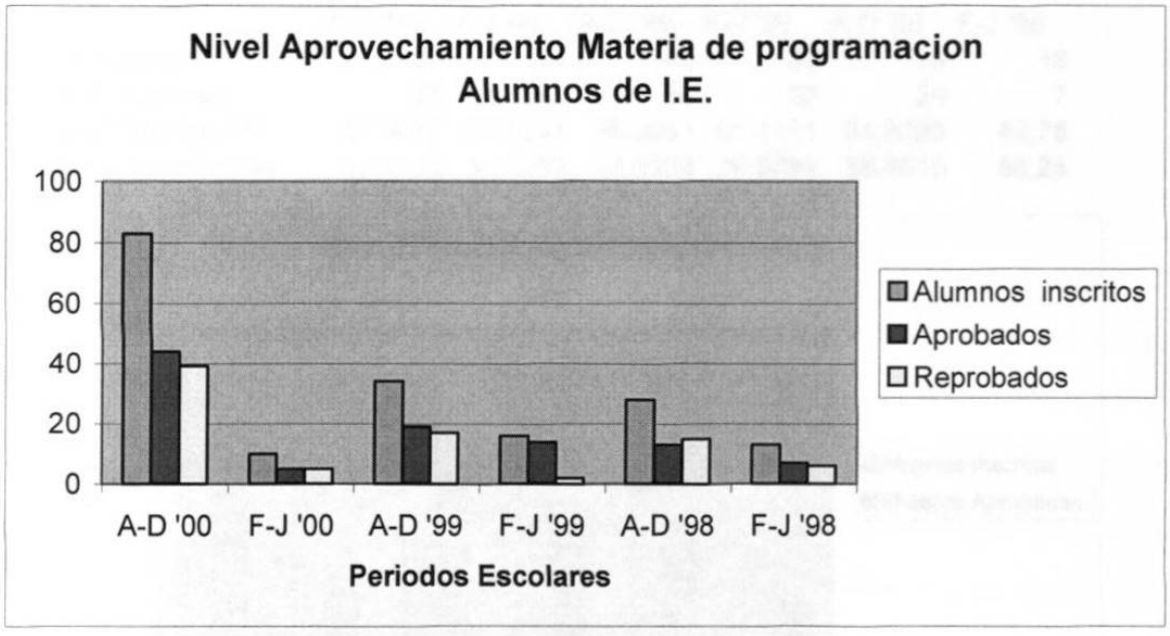


Fig. 12 Comparativo de nivel de aprovechamiento en programación I.S.C. y I.E.

Calculando un Índice de Reprobación Promedio para ambas carreras se tiene:

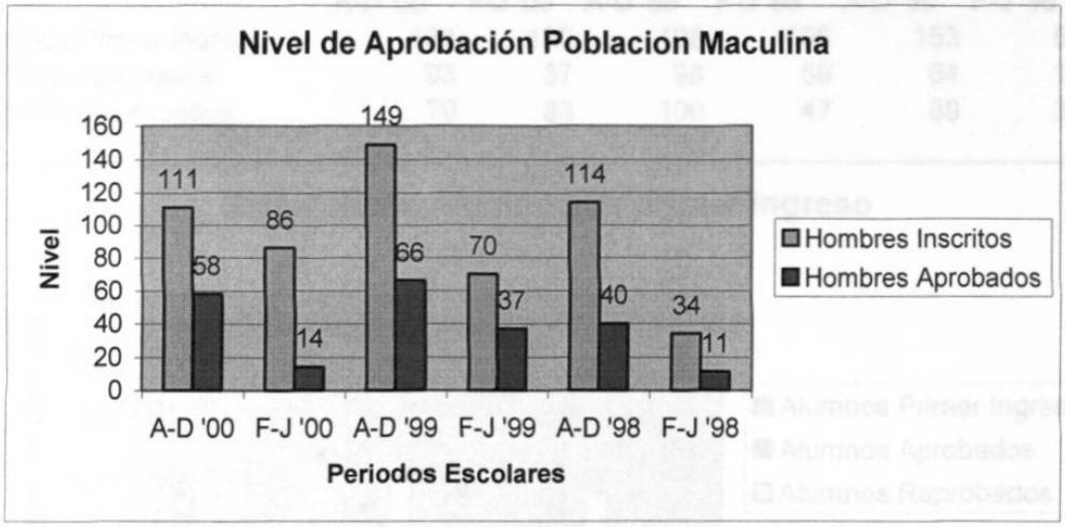
Ingeniería Electrónica 43.20

Ingeniería en Sistemas Computacionales 30.97

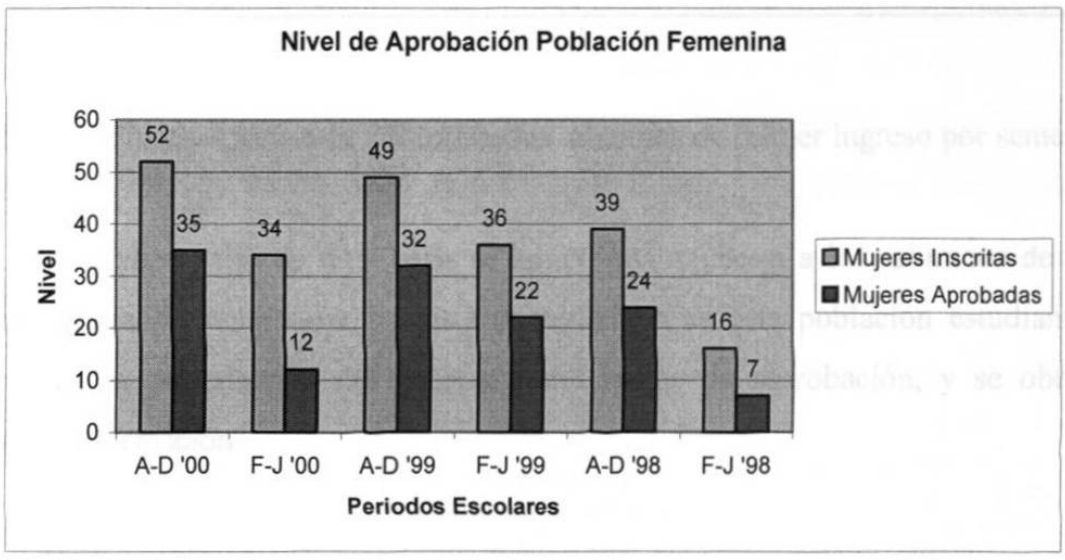
4.5 Comparativo de Aprovechamiento entre hombres y mujeres.

Información adicional es un análisis comparativo de niveles de aprovechamiento entre hombres y mujeres, el análisis nos refleja lo siguiente:

	A-D '00	F-J '00	A-D '99	F-J '99	A-D '98	F-J '98
Hombres Inscritos	111	86	149	70	114	34
Hombres Aprobados	58	14	66	37	40	11
% Hombres Aprobados	52,2523	16,2791	44,2953	52,8571	35,0877	32,3529
% Hombres Reprobados	47,7477	83,7209	55,7047	47,1429	64,9123	67,6471



	A-D '00	F-J '00	A-D '99	F-J '99	A-D '98	F-J '98
Mujeres Inscritas	52	34	49	36	39	16
Mujeres Aprobadas	35	12	32	22	24	7
% Mujeres Aprobadas	67,3077	35,2941	65,3061	61,1111	61,5385	43,75
% Mujeres Reprobadas	32,6923	64,7059	34,6939	38,8889	38,4615	56,25



Proporcionalmente aprueban la materia mas mujeres que hombres.

Fig. 13 Comparativo de aprovechamiento entre hombres y mujeres

4.6 Características de Población estudiantil de Primer Ingreso

Tema de preocupación en el Instituto Tecnológico de Nuevo León es Índice de Reprobación de los alumnos en la materia "Diseño Estructurado de Algoritmos", lo cual se ha mencionado en el punto 4.3, donde se muestra la siguiente gráfica:

	A-D '00	F-J '00	A-D '99	F-J '99	A-D '98	F-J '98
Alumnos Primer Ingreso	163	120	198	106	153	50
Alumnos Aprobados	93	37	98	59	64	18
Alumnos Reprobados	70	83	100	47	89	32

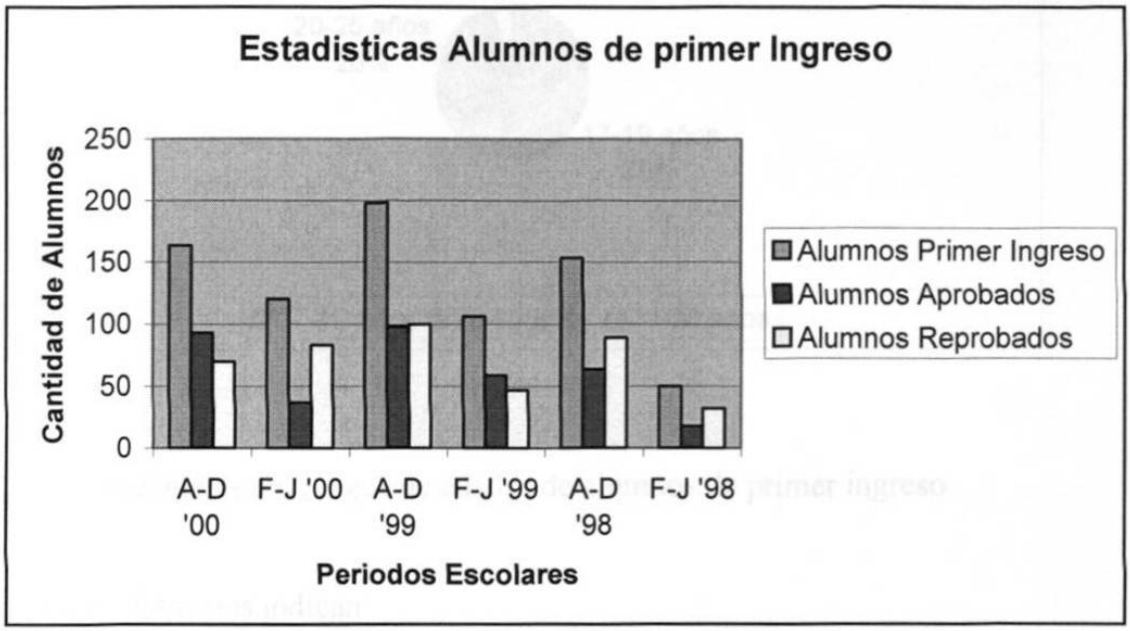


Fig. 14 Estadística Aprobados y Reprobados alumnos de primer ingreso por semestres

Para el desarrollo de ésta Tesis se aplicó una encuesta a los alumnos de primer ingreso, buscando determinar rasgos característicos de ésta población estudiantil que incidan en la presentación del problema del índice de reprobación, y se obtuvo la siguiente información.

4.6.1 Edades.

17-19 años	79
20-25 años	30
25-30 años	5

Fig. 15 Porcentajes en rangos de edades de alumnos de primer ingreso

Estos datos nos indican

- Que la mayor parte de alumnos que ingresan al I.T.N.L. son estudiantes muy jóvenes, recién egresados de la Preparatoria (ó Bachillerato).
- También ingresan alumnos que por alguna razón abandonaron sus estudios por un tiempo, y después regresan a darles continuidad .

4.6.2 Porcentajes de hombres y mujeres.

Hombres	78
Mujeres	36

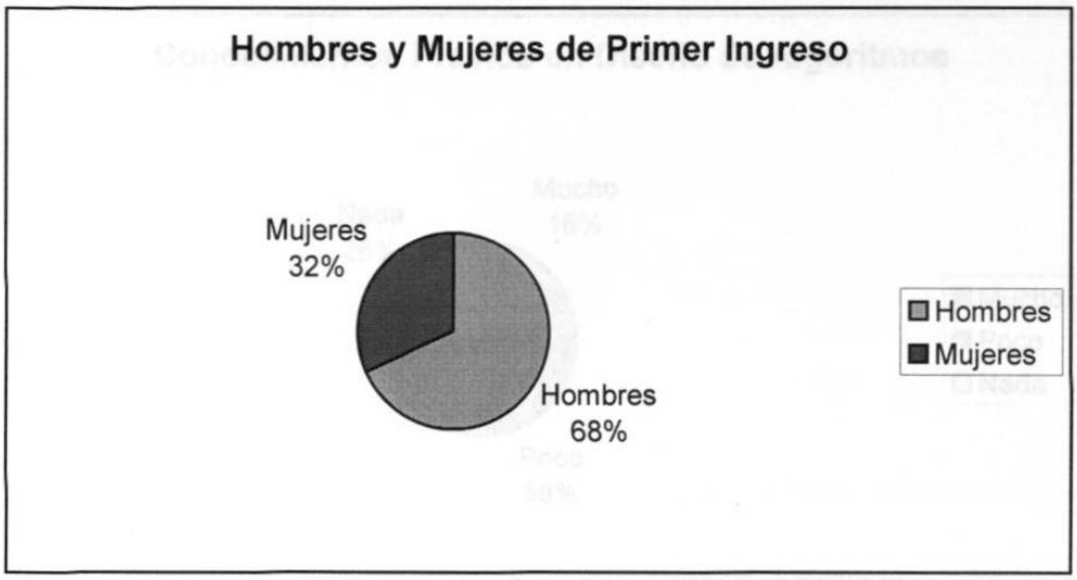


Fig. 16 Porcentajes de hombres y mujeres en alumnos de primer ingreso

Estos datos nos muestran:

- Que la carrera de Ingeniería en Sistemas Computacionales por de rama ingenieril tiene mas aceptación entre los alumnos de sexo masculino.

4.6.3 Conocimientos previos en Diseño de Algoritmos.

Mucho	18
Poco	66
Nada	30

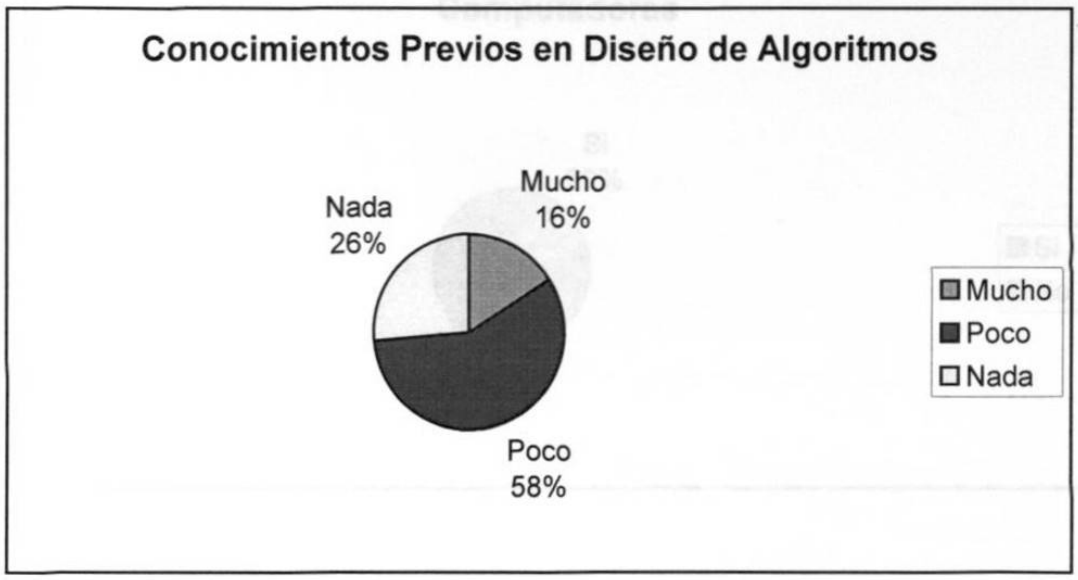


Fig. 17 Conocimientos previos en Diseño de Algoritmos en alumnos de primer ingreso

Estos datos de la encuesta nos indican:

- Que algunas Instituciones de nivel educativo precedente ya han incluido programas de estudio que realizan importante aportación en el estudiante sobre conocimientos de técnicas para el diseño de algoritmos de solución a problemas específicos.
- Sin embargo el mayor porcentaje de los alumnos reconocen que sus conocimientos al respecto son "poco" y "nada", esto nos indica que ingresan al I.T.N.L. con cierta "debilidad" en el diseño de Algoritmos.

4.6.4 Conocimientos previos en Programación.

Si	25
No	89

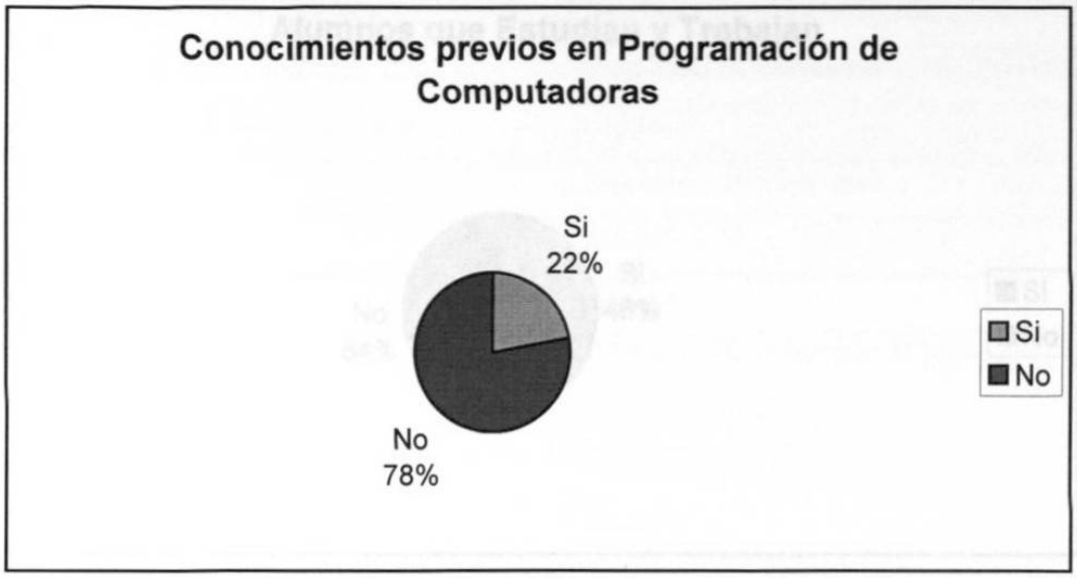


Fig. 18 Conocimientos previos en programación de alumnos de Primer ingreso

Estos datos nos muestran:

- Que algunos alumnos al momento de ingresar al I.T.N.L. ya conocen ó ya tienen alguna experiencia en Programación de computadoras, pero son una mínima población de primer ingreso.
- La mayor parte de la población de primer ingreso no han tenido un acercamiento al conocimiento de algún lenguaje de programación.

La encuesta también incluía una pregunta donde se les pedía a estos alumnos que mencionaran los lenguajes de programación en los cuales ya tenían alguna experiencia, y mencionaron algunos como:

- Power Builder, Turbo C, C++, Turbo Pascal, Visual Fox Pro, Dbase III, Dbase IV, Qbasic, Visual Basic, Access, HTML.

4.6.5 Alumnos que estudian y trabajan.

Si	52
No	62

Fig. 19 Porcentaje de alumnos de Primer ingreso que estudian y trabajan

Estos datos nos muestran que:

- Casi la mitad de la población estudiantil de primer ingreso son alumnos que "ya" tienen un empleo y desean combinar las actividades de "Estudio y Trabajo", pero esta combinación de actividades les genera un grave problema que es la administración del tiempo; ya que esto les reduce grandemente su tiempo disponible para la realización de tareas así como de tiempo de estudio fuera de las aulas de clase.

4.6.6 Tipos de empleos de los Alumnos que trabajan.

Relacionados con el Área de Sistemas	Sin Relación con el Área de Sistemas
Asesor de Computación	Soldador
Auxiliar de Soporte Técnico	Vendedor en Supermercado
Soporte Técnico a Redes	Empleado de Ferretería
Programador de Computadoras	Cajero
Capturista	Empleado de Abarrotes
	Músico
	Taxista
	Vendedor de Productos al Menudeo
	Coordinador de ventas
	Técnico Radiólogo
	Mesero
	Promotor de ventas Teléfonos Celulares
	Auxiliar Contable
	Auxiliar Administrativo
	Dueño de Negocio Propio
	Supervisor de Control de Calidad
	Empleado Empresa de Seguros
	Empleado de Carnicería
	Empleado en una Imprenta
	Instalador de Ductos de Aire
	Maestro de Primaria
	Chofer de Camión
	Electricista

Tabla 1 Tipos de empleos de los alumnos de primer ingreso que trabajan

Estos datos nos muestran que :

- La mayoría de los empleos de los alumnos de primer ingreso no tienen relación con el área de Sistemas, pero ellos tienen el deseo de a través de sus estudios relacionarse con ésta área.

4.6.7 Porqué elegir la profesión de I.S.C.

La encuesta realizada incluyó también que el alumno contestara a la siguiente pregunta ¿Porqué elegiste estudiar la carrera de Ingeniería en Sistemas Computacionales?, y sus respuestas se englobaron en las siguientes:

- Soy Técnico en Computación y deseo mayor preparación.
- Porque el área computacional tiene mucha demanda y quiero superarme.
- Porque me gusta todo lo relacionado con la computadora.
- Porque me gusta mucho hacer programas para la computadora.
- Porque es un reto para mi.
- Porque deseo conocer el funcionamiento interno de la computadora.
- Porque esta carrera tiene mucho futuro.
- Porque esta profesión es muy bien pagada.
- Porque deseo poner mi propio negocio de computación.

4.7 Conclusiones de la encuesta a alumnos de Primer Ingreso.

En la revisión de los datos de la encuesta realizada se obtuvo la siguiente información:

- La mayoría de los alumnos son jóvenes de entre 17 y 19 años, recién egresados de la preparatoria.
- La gran mayoría son alumnos de sexo masculino, solo un pequeño porcentaje son de sexo femenino.
- La mayoría de los alumnos al ingresar tienen gran dificultad al aplicar el método analítico en la solución de problemas, orientado a la programación.
- Una minoría de alumnos "ya" conocen código y sintaxis de uno ó varios lenguajes de programación, algunos de ellos ya conocen y aplican lenguajes de programación de tipo visual (RAD), pero algunos de ellos son solo "codificadores" ya que no han ejercitado el Análisis del problema y su solución algorítmica.
- Casi la mitad de los alumnos "ya" tienen un empleo de jornada de trabajo completa(8 horas/dia).
- La mayoría de los alumnos que trabajan tienen empleos NO relacionados con el área de Sistemas.

4.8 Factores que inciden en el Indice de Reprobación en la Materia de Diseño Estructurado de Algoritmos.

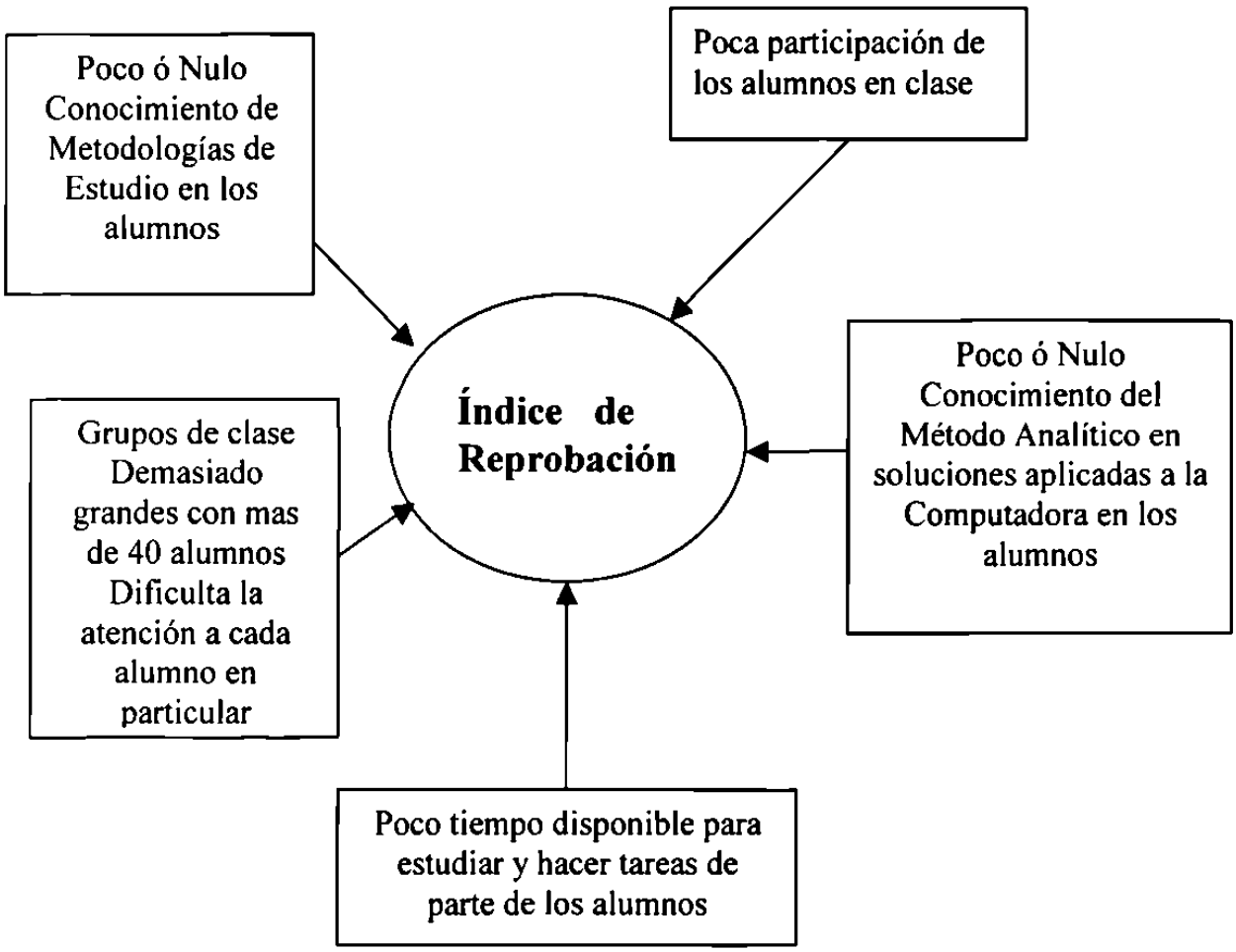


Fig. 20 Factores que inciden en el índice de reprobación de la materia de Diseño Estructurado de Algoritmos.

4.9 Propuesta para reducir el índice de reprobación en la materia de Diseño Estructurado de Algoritmos

- En los cursos propedéuticos impartir como primer tema "Metodologías de estudio", el cual debe concluir al inicio de los demás temas a saber: matemáticas, Química, etc.., de forma que los alumnos puedan aplicar un método de estudio desde el inicio de estos cursos.
- En los cursos propedéuticos incluir el tema "Lógica de programación" que inicie en los alumnos el uso del método analítico para la resolución de problemas, orientado a la programación de computadoras. Esto ayudará no solo a los futuros alumnos de la carrera de Ingenieria en Sistemas Computacionales sino también a los futuros alumnos de las demás carreras de corte ingenieril, ya que ellos también cursarán la materia de Programación.

5

EL PRIMER LENGUAJE DE PROGRAMACION

La programación de computadoras nació con la misma computadora, con las primeras computadoras la programación fue sumamente difícil ya que se realizaba en lenguaje de máquina absoluto a través de conexiones de cables específicos para controlar las funciones elementales de la máquina. En la década de 1950 el proceso mejoró con la utilización de tarjetas perforadas ya que con ello se logró transferir la escritura de programas a las tarjetas y su lectura automática en vez de insertar cables en el hardware, aunque los procesos de compilación y ejecución seguían requiriendo esperar que no se fundiese algún bulbo que abortara el proceso, por ese tiempo no se disponían de sistemas operativos y normalmente el mismo personal construía, programaba y daba mantenimiento a la computadora.

La aparición del transistor y su aplicación en las computadoras logró hacerlas mas confiables así como su comercialización, esto coadyuvó a una separación y especialización de funciones, ya que el mismo equipo de personas que construía la computadora no podría "ir" a programar y dar mantenimiento a todas las computadoras vendidas, de forma que las funciones "programar" y dar "mantenimiento" debían de hacerlas otros equipos de personas.

El "ambiente de programación" tiene que ver con las herramientas que faciliten al programador realizar su función, a este respecto se puede afirmar que la programación de computadoras ha ido de la mano con la evolución de los sistemas operativos.

5.1 Ambientes de Programación

El ambiente de desarrollo "batch" también es conocido como "Sistema de procesamiento por lotes " fue la manera de operar de los computadores de la primera y segunda generación que comprende de inicio 1951 hasta 1964, en estas generaciones no habían sistemas operativos, y las empresas fabricantes de computadoras producían también el software a usarse en ellas, estas computadoras solo podían ejecutar un proceso a la vez, y las características del "ambiente de programación" fueron las siguientes:

a) El programador realizaba la codificación del programa en hojas de papel graficadas con el formato correspondiente al lenguaje en uso(Fortran, Cobol, Assembler, etc..)
b) Pasaba la codificación a tarjetas perforadas
c) Llevaba las tarjetas perforadas a la sala de lectura y las dejaba con el operador de computadora para la lectura y compilación del programa
d) Esperaba a que el operador le entregara un listado emitido por la computadora dando los resultados de la compilación
e) Si había errores se hacían las correcciones y se realizaba de nuevo el proceso a partir del paso b)

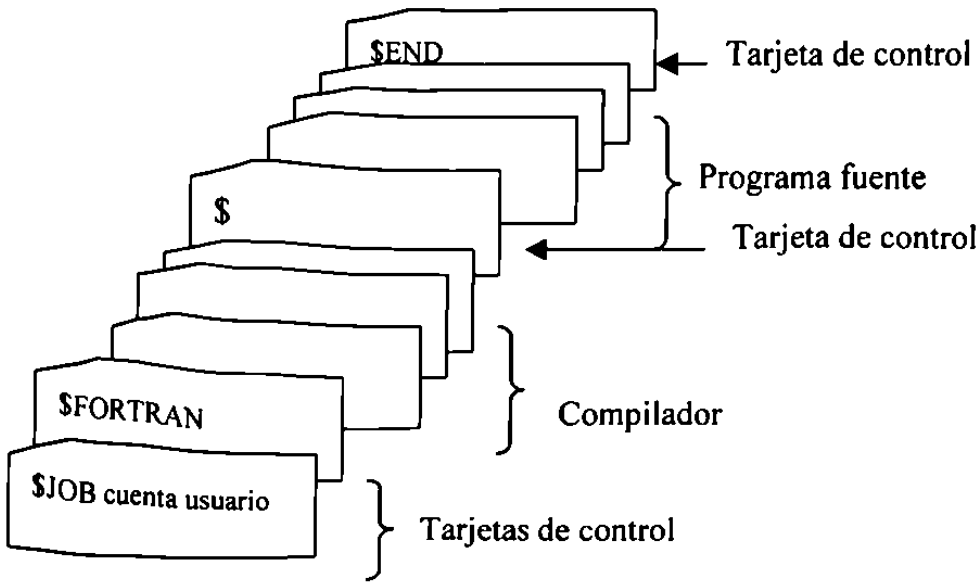


Fig. 21 Estructura del lote de un proceso de compilación

Las computadoras de la tercera generación tuvieron la característica de ser construidas en base a circuitos integrados(a baja escala), tecnología que les proporcionó ventajas en precio y desempeño además de gran poder de computo, ya que podían realizar procesos comerciales y científicos.

En esta generación (1964 – 1981) aparecen los primeros sistemas operativos y con ellos las técnicas de partición de memoria y multiprogramación, que dieron lugar a llamarlas "computadoras multiusuario", esto les permitieron atender a varios usuarios en forma simultánea, cada usuario operaba una terminal conformada por una consola y un teclado conectados con la computadora en forma local ó remota, el uso de la consola por parte del usuario hizo posible el surgimiento del término "interactivo".

Con los sistemas operativos aparecieron también los editores de texto que permitieron a los programadores la captura de sus programas y grabarlos como archivos en el disco magnético, donde también residían los compiladores, este "ambiente de programación y Procesos de datos" dejó obsoleto el uso de las tarjetas perforadas.

Como "ambiente de programación" la característica principal fue que tanto el editor como también el compilador y sus Librerias eran herramientas que no estaban integradas, y el programador operaba con ellas en forma separada. Varios programadores podían trabajar al mismo tiempo, operando cada uno su propia terminal, y también compilando sus programas al mismo tiempo.

Los procesos de compilación y de datos seguían siendo "batch", y su ejecución se "ordenaba" mediante un archivo "Job Control" también capturado en el editor de texto, que contenía las órdenes mediante comandos dirigidos al sistema operativo(similar a los archivos .bat operados en MS-DOS), que le indicaban al computador las actividades a realizar , en detalle paso a paso.

En las computadoras de esta generación seguía persistiendo el hecho de que las empresas que fabricaban el hardware también producían el software que se usaría en ellas.

Las actividades del proceso de compilación eran las siguientes:

a) El programador capturaba la codificación del programa en el editor de texto dando al archivo el formato requerido de acuerdo al lenguaje a usar(Cobol, Fortran, Assembler,etc...)
b) El programador capturaba el archivo Job Control en el editor de texto
c) El programador enviaba a ejecución el archivo Job Control que entraba a una cola de espera
d) El programador esperaba el resultado que arrojaba la compilación mediante un listado de la compilación que indicaba si había o nó errores, el listado de la compilación podía ser visto en pantalla a través del editor ó enviarlo a la impresora.
e) Si había errores se hacían las correcciones en el archivo correspondiente y se repetía el proceso a partir del paso c)

EJEMPLO DE UN ARCHIVO "JOB CONTROL"

```
$JOB  cuenta usuario
$lenguaje a usar
$CALL compilador
$CALL nombre del programa fuente
$END
```

Fig. 22 Segmento de un archivo Job Control

La aparición de la microcomputadora marcó el inicio de las computadoras de cuarta generación y también un parte aguas en la generación de software, ahora además del software de los diseñadores y fabricantes de hardware(IBM, Hewlett Packard, Digital Equipment, etc.) aparecieron nuevas empresas desarrolladoras de software(Microsoft, Borland Inc., Lotus Development Corp., etc..) cuyos productos pueden ejecutarse en cualquier microcomputadora que sea de arquitectura abierta (IBM y compatibles).

En el ámbito de las microcomputadoras se generó una rápida evolución de Sistemas Operativos (y con ello el "ambiente de programación"), renglón en el cual ha dominado le empresa Microsoft quien con diversas versiones del Sistema Operativo MS-DOS predominó en el mercado de las microcomputadoras.

La empresa Apple Computer fue la primera en fabricar hardware y software para microcomputadoras, y su política de mercadeo fue apartarse de la arquitectura abierta impulsada por IBM y Microsoft Co., de tal forma que hasta la actualidad su arquitectura sigue siendo "NO compatible", sin embargo hasta la actualidad se ha mantenido en el mercado con su singular computadora modelo Macintosh.

El "ambiente de programación" imperante durante el dominio del Sistema Operativo MS-DOS fue similar al "ambiente de consola" de las computadoras de la tercera generación, solo que las microcomputadoras solo pueden atender a un usuario a la vez. Para atender a varios usuarios a la vez usando microcomputadoras varias empresas desarrollaron tecnología para comunicaciones entre computadoras (entre ellas Novell Inc.), dando lugar al ambiente de Redes LAN.

En 1983 Borland Internacional Inc. Puso a disposición del mercado computacional un compilador del lenguaje Pascal y lo llamó "Turbo Pascal", el cual tenía las siguientes características:

- Eficiente y rápido
- Sumamente económico ($49.95 dls.)

- Ambiente de programación "Turbo" (Editor, Compilador, Librerias y Depurador integrados como un solo programa)

este producto tuvo una gran aceptación no solo en ambientes académicos sino también los demás sectores de la computación y dio inicio al "ambiente de programación" llamado "Turbo", para el cual posteriormente también se dispuso de otros productos como "Turbo Basic", "Turbo C", etc..

La programación en ambiente MS-DOS también se le conoció como "programación de 16 bits" por la capacidad de direccionamiento en su arquitectura.

En 1984 la empresa Apple Computer anunció comercialmente su modelo Macintosh que ofrecía el primer sistema operativo con ambiente gráfico operado mediante "ventanas" y de un dispositivo denominado "mouse", a este ambiente se le denominó "Ambiente GUI" (Graphical User Interface).

Esto representó un reto para la empresa Microsoft de manera que en marzo de 1987 anunció su primera versión de sus sistema operativo "Windows 1.0", pero no tuvo el éxito que se esperaba, y los usuarios preferían seguir usando el MS-DOS. Microsoft siguió insistiendo hasta que en 1990 anuncia la versión "Windows 3.0" con la cual logró un éxito sin precedentes, y con esta versión de "Windows" la mayoría de los usuarios tanto "caseros" como de "negocios" migraron del ambiente de MS-DOS al "Ambiente Windows".

En estas versiones del sistema operativo aparecieron las interfases de programación API(Application Programming Interface) para versiones Windows 3.x.

En 1995 Microsoft promueve en el mercado su nueva versión de Windows denominado "Windows 95" con cambios muy significativos, y posteriormente en 1998 se reafirmó con una nueva versión llamada "Windows 98".

Los cambios sustanciales fueron :

- Programación a 32 bits
- Programación RAD (Rapid Applications Development)
- Dispone de una gran cantidad de API y DLL's que facilitan el trabajo de la programación
- Permite la programación tipo "Visual"
- Ambiente de programación IDE (Intergrated Development Environment), ó "Ambiente Integrado de Desarrollo" cuya característica es que contiene: un Editor de código, un Depurador de errores, una barra de herramientas visuales, una Barra de herramientas de manejo de Bases de datos, un Formulario para diseño de ventanas y un Inspector de Objetos.

5.2 El Modelo Cliente/Servidor

Los avances en la tecnología de comunicaciones entre computadoras ha permitido el amplio uso del Redes de computadoras permitiendo la distribución de los recursos de cómputo, este ambiente de redes ha desplazado poco a poco al "Modelo Host" ya que éste tiene como característica principal la centralización de procesos, en cambio el ambiente de Red permite la descentralización de los mismos.

La aparición de la microcomputadora ha dispuesto capacidad de procesamiento de datos fuera de los Centros de Cómputo, esta nueva relación cambia el paradigma de Cómputo centralizado y genera un nuevo paradigma llamado "Cliente/Servidor", el cual tiene implicaciones tanto de hardware como de software.

En cuanto a software el término "Cliente/Servidor" define una relación entre dos programas de computación, donde uno de ellos(el Cliente) solicita un servicio, y el otro (el Servidor) satisface la solicitud. El Modelo "Cliente/Servidor" brinda una forma conveniente y simple para interconectar programas que están distribuidos en locaciones distantes. En este Modelo un "Servidor" que a veces se denomina "daemon" ó "demonio" es activado, y permanecerá en espera de alguna petición ó solicitud de algún "Cliente", normalmente muchos programas "Cliente" podrán compartir los servicios potenciales de un "Servidor" común.

El Modelo "Cliente/Servidor" requiere disponer de varios elementos como: Arquitectura completa de Telecomunicaciones, Administración de archivos de respaldo, Control de Procesos distribuidos, Mensajería, Comunicación entre Aplicaciones, Bases de datos Distribuidas, Lenguajes de consulta, Bases de datos independientes de los Lenguajes de consulta, Aplicaciones independientes de las Bases de datos y de los tipos de terminales.

Los componentes principales de una aplicación "Cliente/Servidor" son consideradas tres, que son:

- Interfase al usuario (presentación)
- Administración de los datos
- Procesamiento de los datos

estos componentes pueden estar distribuidos a lo largo de toda la red, éste Modelo implica soporte de ambientes complejos de múltiples plataformas y protocolos, pero reteniendo la simplicidad de un sistema individual.

En el acceso remoto de datos la Administración de los mismos se lleva a cabo el el "servidor" y el procesamiento y presentación de los datos se lleva a cabo en el "Cliente".

El uso del Modelo "Cliente/Servidor" se ha convertido en una de las características principales de los Lenguajes de programación, ya que la gran mayoría de las aplicaciones de negocios utilizan este modelo. Los programadores en ambiente "Cliente/Servidor" deben asumir responsabilidades mayores a las que se tenían en el pasado, facilitando en sus aplicaciones la toma de decisiones operativas, tácticas y estratégicas.

5.3 Metodologías de la Programación

Los tiempos de la programación a través de interruptores eléctricos, de tiras de papel perforada y listas de bytes en hexadecimal han pasado, primero se avanzó hacia los lenguajes de alto nivel como Cobol, Fortran, Algol entre otros con la Programación Estructurada y la Programación Modular, el siguiente paso parecen ser los ahora populares lenguajes de programación visual.

Los lenguajes visuales ostentan este nombre porque proporcionan al programador de herramientas gráficas que le permiten mejorar la velocidad en el desarrollo de un proyecto, pues simplifican en gran medida la construcción de la interfaz con el usuario, lo cual representa generalmente mas del 50% del trabajo que requiere un proyecto típico. No se puede concebir la programación actual con los lenguajes de hace una década, sino con los lenguajes de hace 3 ó 4 años, muchas herramientas de desarrollo en la actualidad se han incorporado al nuevo paradigma de la programación visual.

La mayor ventaja del desarrollo visual es que permite generar aplicaciones funcionales, sencillas y a la vez elegantes en forma rápida.

Programar computadoras en la actualidad no solo es generar código para procesarlo a las mas altas velocidades posibles, ahora se requiere programar la parte medular de cualquier software(el procesamiento de la información) cuidando la interfaz con el usuario, y la generación de ayuda en línea(mediante el paradigma del hipertexto para moverse con facilidad sobre información), sin dejar de lado la obligatoriedad de un código rápido y eficiente entre otras cosas.

** Notas extraídas del articulo "El universo de los lenguajes visuales", de los autores Alejandro Arcos, David Garza, Manuel López, Marco A. Pérez.
Article A20504689 1997-Servicios Editoriales Sayrols S.A. de C.V.

El medio ambiente computacional y las metodologías de la programación han cambiado y se han dirigido hacia el desarrollo de aplicaciones Cliente / servidor y de interfase gráfica(GUI), esto representa un cambio en el paradigma de la programación ya que las aplicaciones son "Orientadas al usuario(Visuales)" en vez de "Orientadas al Programador" y al los Sistemas Gerenciales Integrados(MIS).

Los 4GL's están orientados a incrementar la velocidad del desarrollo de las aplicaciones y a satisfacer las necesidades de los usuarios de los Sistemas denominados "desk System" ó sistemas de escritorio con calidad y consistencia, así como facilidades en el diseño y modelado de datos integrados con herramientas CASE, de forma que el Programador pueda dedicarse al diseño y optimización de la solución.

La Programación Visual tiene implícita el uso de 2 Metodologías a saber:

- Programación Orientada a Objetos
- Programación Orientada a Eventos

y a los lenguaje de Programación Visual se les considera como "Lenguajes de cuarta generación(4GL's)

La Programación Orientada a eventos tiene como característica el uso de componentes(objetos) que le dan al usuario la posibilidad de construir aplicaciones por medio de interfaces gráficas en base a la ocurrencia de "Eventos". Los programas "Orientados a Eventos" son aplicaciones típicas del manejo del ambiente Windows que una vez que han iniciado su ejecución quedan en espera de las acciones de los usuarios que en este caso son llamadas "Eventos".

En la Programación Orientada a Eventos cada componente u objeto tiene definidas "Propiedades" y es manipulado por "Métodos", los eventos típicos son por ejemplo "un click sobre un botón", y se activa un "Método" que puede ser para Abrir un Archivo, ó escribir sobre una caja de Texto ó activar una nueva ventana, etc..

Por definición los Lenguajes de cuarta generación(4GL's) son los lenguajes de "Programación Asistida" por medio de ayudantes ó "Wizard", estos lenguajes se han diseñado para facilitar la realización de muy variadas tareas, tanto de tipo científico, estadístico y de negocios.

A muchos de los Lenguajes 3GL les han sido implementadas estas nuevas características y han evolucionado como 4GL's, entre ellos podemos encontrar : Visual Basic, Visual C++, Visual J++, INFORMIX 4GL, Delphi(Pascal), etc...

Los 4GL's contienen facilidades para el acceso a Bases de datos tanto de tipo Relacional como también a Bases de datos Orientados a Objetos.

5.4 Lenguajes Visuales (4GL's) de propósito general mas usados actualmente

Lenguaje Visual Basic

Es un producto de la casa de software **Microsoft**, es una herramienta RAD(Rapid Appication Develpment), éste lenguaje fue el iniciador del paradigma del desarrollo visual para Windows y desde su aparición en 1991 ha demostrado facilidad de uso y de aprendizaje, sus características de desarrollo han mejorado a través de sus múltiples versiones incrementando de esta forma su poderío y eficiencia.

Su ambiente de programación es de tipo IDE y permite generar aplicaciones de negocios, con Bases de datos e Internet. Tiene capacidad para creación de controles ActiveX y un compilador de código nativo, pero no puede crear ejecutables por sí solo ya que requiere de por lo menos un archivo DLL para que funcione su ejecutable. Puede generar aplicaciones Cliente / servidor y cuenta con asistentes de ayuda Intellisense. Soporte la Programación Orientada a Objetos y puede ser usado como lenguaje de para otros productos de Microsoft Office(Word, Excell, Access).

Es compatible con proveedores nativos de OLE DB para operar con MS SQL Server y Oracle 7.3.3+. La versión mas actualizada a la fecha es Visual Basic 6.0 y se comercializa en tres presentaciones: Empresarial, Profesional y Para la Enseñanza, la versión para la enseñanza es la mas austera mientras que la Empresarial es la mas sofisticada.

Este lenguaje es una buena opción para el desarrollo de aplicaciones en Windows 95, Windows 98 , Windows NT, y Windows 2000.

5.4 Lenguajes Visuales (4GL's) de propósito general mas usados actualmente

Lenguaje Visual Basic

Es un producto de la casa de software **Microsoft**, es una herramienta RAD(Rapid Appication Develpment), éste lenguaje fue el iniciador del paradigma del desarrollo visual para Windows y desde su aparición en 1991 ha demostrado facilidad de uso y de aprendizaje, sus características de desarrollo han mejorado a través de sus múltiples versiones incrementando de esta forma su poderío y eficiencia.

Su ambiente de programación es de tipo IDE y permite generar aplicaciones de negocios, con Bases de datos e Internet. Tiene capacidad para creación de controles ActiveX y un compilador de código nativo, pero no puede crear ejecutables por sí solo ya que requiere de por lo menos un archivo DLL para que funcione su ejecutable. Puede generar aplicaciones Cliente / servidor y cuenta con asistentes de ayuda Intellisense. Soporte la Programación Orientada a Objetos y puede ser usado como lenguaje de para otros productos de Microsoft Office(Word, Excell, Access).

Es compatible con proveedores nativos de OLE DB para operar con MS SQL Server y Oracle 7.3.3+. La versión mas actualizada a la fecha es Visual Basic 6.0 y se comercializa en tres presentaciones: Empresarial, Profesional y Para la Enseñanza, la versión para la enseñanza es la mas austera mientras que la Empresarial es la mas sofisticada.

Este lenguaje es una buena opción para el desarrollo de aplicaciones en Windows 95, Windows 98 , Windows NT, y Windows 2000.

Algunas características propias de Visual Basic 6.0

- Compilador de código nativo de alto rendimiento construyendo aplicaciones Cliente/servidor optimizado con el compilador optimizado de Visual C++
- Acceso a todas las fuentes de datos corporativas mediante ODBC, OLE DB y Microsoft ActiveX Data Objects(ADO), mediante controladores incorporados para SQL Server, Oracle, Microsoft Access, ODBC y SNA Server
- Herramientas de Bases de datos Integradas
- Enlace de datos automático ajustando propiedades en la ventana de propiedades y lo conecta a cualquier fuente de datos
- Facilidad en la creación de componentes COM
- Diseñador Data Environment para generar en forma visual objetos reutilizables de tipo recordset enlazando múltiples fuentes de datos para adición y manipulación
- Diseñador Web Class para permitir la creación de componentes de Servidor accesibles desde cualquier navegador sobre cualquier plataforma
- Aplicaciones de 32 bits

Tabla 2 Algunas características entre versiones de Visual Basic 6.0

Características	Empresarial	Profesional	Enseñanza
Uso de Visual Basic en Aplicaciones de Microsoft Office(Word,Excell,Access)	si	si	si
Reducción en tiempo de desarrollo usando controles explorador Web, Check Box, etc..	si	si	no
Construcción de componentes en código nativo	si	si	no
Generación de Aplicaciones de Internet en cualquier plataforma	si	si	no
ADO(ActiveX Data Objetc) en Redes LAN	si	si	no
Microsoft Visual SourceSafe sistema para control de versiones para compartir y asegurar código en desarrollo para Windows y la Web	si	no	no
Microsoft Data Engine(MSDE) para Visual estudio 6.0 para soluciones compartidas compatibles con SQL 7.0 y migración a SQL Server	si	no	no
Data Objetc Wizard para simplificar la creación de componentes COM	si	no	no
Creación de fuentes de datos personalizadas	si	si	no
Creación de formularios e informes con drag & drop	si	si	no
Soporte nativo de procesadores Alpha para plataformas DEC Alpha	si	no	no
Depurador de sentencias SQL(TSQL) para depuración interactiva de procedimientos almacenados y Trigger´s	si	no	no
Desarrollo de Aplicaciones de 32 bits	si	si	si
Manejo de punteros en forma natural	no	no	no
Soporte a aplicaciones de consola	no	no	no

Lenguaje Delphi

Es un producto de la casa de software Borland, es una herramienta RAD(Rapid Application Develpment) adecuada para crear aplicaciones distribuidas de Internet y de Bases de datos, la versión mas reciente es Delphi 5 y a ella le han sido añadidas una larga lista de funciones nuevas para desarrolladores individuales y para equipo de desarrollo.

Delphi permite generar aplicaciones de 32 bits y gestionar todo tipo de datos tanto multimedia como gráficos, sonidos y animaciones. Asimismo es posible manejar Bases de datos de múltiples formatos, así como descender al mínimo detalle del computador hasta accesar directamente a periféricos, dispositivos de entrada y de salida é incluso puntos de pantalla.

Delphi está basado en el lenguaje Object Pascal lo cual le permite operar en forma natural la programación Orientada a Objetos y también de punteros, éste lenguaje se distribuye en 3 presentaciones:

- Delphi Enterprise
- Delphi Professional
- Delphi Standard

Delphi Standard es una versión austera propia para la enseñanza de este lenguaje, mientras que la versión Enterprise es la versión mas sofisticada con la totalidad de características propias de Dephi 5.

Algunas características propias de Delphi 5

- Aplicaciones compiladas en código nativo, no requiere intérprete Run Time
- Robustez con manejador de excepciones
- Drives de Bases de datos nativas de alta velocidad, libres de regalías para : Paradox, Access, Fox Pro, Dbase, AS/400, DB2, MS SQL Sever, Oracle, Informix e InterBase
- Herramientas Cliente / servidor como: Monitor SQL, Explorador SQL, Asistente de migración a Bases de datos
- Cubo de decisión para soporte de decisiones y análisis de negocios
- Reporteador integrado, análisis gráficos y multidimensionales
- Implementación nativa de COM en estándares industriales
- Implementación nativa de CORBA
- Capacidad en Bases de datos escalables
- Diccionario de datos escalable para consistencia visual e integridad de datos
- Tecnología de compilación de paquetes para ejecutables mas pequeños
- Arquitectura Orientada a Objetos
- Salidas Internet HTML

Tabla 3 Algunas características entre versiones de Delphi 5

Característica	Enterprise	Professional	Standard
Desarrollo a 32 bits con soporte a Win API	si	si	si
Alto desempeño a 32 bits, con compilador de código nativo	si	si	si
Acceso completo a Win 32 API, ActiveX, Multihlo, OLE, OLE DB, DCOM,	si	si	si
Manuales con Documentación impresa	si	si	no
Documentación en línea	si	si	si
Diseñador con Modulo de datos con árbol y visor de diagrama de datos	si	si	no
Aplicaciones ejemplo para facilitar Aprendizaje	si	si	si
Asistente para Aplicaciones de consola	si	si	si
Aplicaciones de servicio a Windows NT	si	si	no
Editor de código AppBrowser con símbolos de hipervínculos	si	si	no
Depurador entre Procesos para pasar de un Proceso a otro	si	si	no
Depurador de procesos remotos para desarrollo distribuido	si	no	no
Browser Internet Explorer	si	si	si
Soporte a JPEG	si	si	no
Asistente Active Server Object para desarrollo en Servidores ASP	si	no	no
Drives para Access, Fox Pro, Paradox, Dbase y Redes LAN	si	si	no
Conectividad con ODBC	si	si	no

Lenguaje C++ Builder

Este lenguaje es producto de la casa de software Borland, es una herramienta RAD(Rapid Application Development), y permite a los desarrolladores la creación de aplicaciones C++ de alta velocidad, desde aplicaciones científicas, comerciales, de gestión de Bases de datos y de Internet, hasta controladores de dispositivos de entrada y salida de datos así como aplicaciones de consola(Ambiente MS DOS).

El lenguaje C++ Builder cubre las necesidades de los desarrolladores de Software base cuyas preocupaciones son la velocidad del compilador y el soporte a estándares establecidas. C++ Builder por naturaleza soporta la Programación Orientada a Objetos y la programación Estructurada, así como las características de la programación Cliente/servidor.

C++ Builder soporte los estándares de la industria de software como: Oracle 8i Data Base Server Oracle Corporation, Microsoft Fundation Classes(MFC) de Microsoft, Microsoft SQL Server 7 y MTS(Microsoft Transaction Server), las Bibliotecas de Inprise Corporation, Librerías de Objetos de Windows (OWL) y Librería de Componentes Visuales(VCL).

La versión mas actualizada a este momento es la versión: C++ Builder 5, y se distribuye en tres presentaciones a saber:

- C++ Builder 5 Enterprise
- C++ Builder 5 Professional
- C++ Builder 5 Standard

la presentación C++ Builder 5 Standard es la versión mas austera, mientras que la versión C++ Builder 5 Enterprise es la versión mas sofisticada.

Algunas Características propias de C++ Builder 5

- Programación para PC, Cliente / servidor y Multi-Nivel
- Tecnología de desarrollo Visual Bidireccional
- Aplicaciones para gestión de Bases de datos
- Desarrollo rápido de aplicaciones distribuidas con objetos estándar CORBA y COM, para plataformas Unix, Java y Windows
- Soporte a especificaciones estándar de C++ ANSI/ISO
- Entorno Integrado de Desarrollo(IDE), con herramientas visuales bidireccionales y su arquitectura de componentes, facilita todo tipo de programación C++ desde sistemas Back-End hasta desarrollos visuales Front-End, todos en una sola herramienta
- Herramientas Internet que incluyen ActiveForms para construir aplicaciones Web en C++ y WebBroker, así como también para desarrollar aplicaciones CGI, WinCGI, ISAPI y NASPI, y soporte a protocolos HTTP, FTP, SMTP, POP, NNTP, HTML y TCP/IP
- Kit de Desarrollo de servicios de Bases de datos Multi-Nivel(MIDAS) con MIDAS-2, para simplificar el desarrollo, integración y despliegue de clientes de Bases de datos distribuidas
- EZ-COM simplifica el desarrollo en C++ de Clientes COM y creación de controladores ActiveX en un paso para soporte a Data Binding

Tabla 4 Algunas características entre versiones de C++ Builder 5

Características	Enterprise	Professional	Standard
AppBrowser con hipervínculos de símbolo e historia de navegación	si	si	si
Class Explorer para el mapa de la fuente de clases y asistente de creación de miembros	si	si	no
Creación Visual de componentes	si	si	si
Compatible con expresiones regulares de PERL	si	si	si
Herencia y encadenamiento Visual de formas	si	si	si
Repositorio de Objetos para almacenar y reutili-Zar formas, módulos de datos y asistentes	si	si	si
Visor de dependencias de archivos del Project Manager	si	si	si
Exportar archivos Project Make	si	si	si
Asistente para aplicaciones Cliente en Win2000	si	si	si
Asistente para desarrollo de Applets	si	si	no
Asistente para Objetos COM	si	si	no
Asistente para Aplicaciones Windows NT	si	si	no
Asistente para Aplicaciones con MFC	si	si	no
Acceso en Aplicaciones de escritorio para Access, Fox Pro, Paradox y DBase	si	si	no
Conectividad completa con ODBC	si	si	no
Enlaces con SQL, InterBase, Oracle, SyBase, Informix, MS SQL Server y DB2	si	no	no
Herramientas de Desarrollo Integrado con Bases de datos	si	si	no
MIDAS Kit de desarrollo	si	no	no
Soporte a Oracle 8i avanzado	si	no	no
Soporte a aplicaciones de consola	si	si	si

Lenguaje Visual C++

Acerca del Lenguaje C y de C++ desde siempre se ha dicho que es difícil, críptico y que no es posible dominarlo del todo, el manejo de variables puntero siempre ha sido molesto y en ello es muy fácil equivocarse, pero el manejo de estas variables es quizá la herramienta mas poderosa en el desarrollo de aplicaciones realmente de corte profesional. Desde su aparición el Lenguaje C++ ha sido dirigido hacia la aplicación de la Programación Orientada a Objetos, y al agregarle las técnicas de la Programación Orientada a Eventos se ha generado el conocido Visual C++, este ha sido utilizado para el desarrollo de programas como Word, Excel, PowerPoint, Fox Pro, Netscape navigator, Internet Explorer, Visual Basic, Paint Shop Pro....y otras mas. Asimismo es usado cuando un lenguaje por sus características no puede con el desarrollo de algo en particular(ejemplo DLL's).

Visual C++ es un producto de la casa de software Microsoft, y para su aprendizaje es necesario también aprender las funciones MFC(Microsoft Fundation Class), el cual es un producto "empacado" junto con Visual C++ y aunque esto incrementa el numero de cosas por aprender al final permiten la creación de aplicaciones empleando "objetos" que ya están probados en su funcionamiento. Se puede decir que Visual C++ es un lenguaje tan difícil como cualquier otro aunque requiere de mas cuidado de parte del programador, pues el manejo de memoria no es automático y esta es la fuente de la mayoría de los errores que se cometen, asimismo es preciso que el programador adquiera mucha práctica para el aprendizaje de los detalles de la MFC.

La versión mas actual a este momento es Visual C++ 6.0, y es distribuido bajo tres presentaciones a saber:

- Visual C++ Enterprise Edition
- Visual C++ Professional Edition
- Visual C++ Standard Edition

Algunas Características propias de Visual C++ 6.0

- Aumenta la velocidad del desarrollo de aplicaciones y simplifica la codificación con la tecnología IntellSense, el uso de wizard's permiten eliminar la memorización de sintaxis complejas, asi como parámetros y propiedades de objeto.
- Los desarrolladores pueden editar código al tiempo que realizan la depuración, sin tener que salir de la sesión, asimismo reestructurar, reiniciar el depurador y regresar a la aplicación donde se generó un problema.
- Integra fácilmente y sin problemas la funcionalidad de Microsoft Word, Microsoft Excel, así como otras aplicaciones con Active Document Containment (Contenido de Documento Activo), los usuarios obtienen las funciones y la capacidad total a que están acostumbrados, mientras que el desarrollador escribe menos código mediante el uso de wizard's.
- Hace que la recompilación sea mas rápida, el tiempo de recompilación se ha reducido en un 30%.
- Dispone de mas de 150 Clases escritas por profesionales, incluidas en la biblioteca (Library) Microsoft Fundation Classes and Templates(MFC&T).
- Soporta las especificaciones ANSI/ISO.
- Permite generar aplicaciones Web basadas en Multimedia altamente interactivas con Dynamic HTML.
- Permite crear controles ActiveX y objetos COM.
- Permite depurar interactivamente procedimientos almacenados en SQL Server.
- Simplifica grandemente el acceso a cualquier puente relacional ó no relacional, correo electrónico y sistemas de archivo, texto y gráficos, objetos de negocio del cliente y mas, con los wizard OLE DB.
- Soporte a aplicaciones de consola

Lenguaje Cobol

El Cobol es un Lenguaje de programación que ha sido capaz de permanecer activo desde el inicio de la informática hasta nuestros dias, y por lo visto no tiene la menor intención de abandonarnos. Hablar de Cobol es remontarnos mas de 40 años atrás, ya que por esa época cada computadora era construida, programada en su propio Sistema Operativo y sus propios Lenguajes de programación, era la época de la guerra fría entre los diversos fabricantes de computadoras y no existía la menor compatibilidad entre computadoras de diferentes marcas fabricantes.

Cobol nació por el deseo de desarrollar un lenguaje de programación que pudiera "correr" en cualquier computadora de cualquier marca, y fue en Mayo de 1959 cuando en los Estados Unidos de América organizó una comisión fabricantes de computadoras , empresas privadas y representantes del gobierno, y la llamó CODASYL(Conference On Data System Languages). Las reuniones de esta comisión dieron origen al Lenguaje Cobol y a su primera versión de le denominó Cobol-60, el lenguaje siguió evolucionando con una nueva versión en 1968 y otra en 1974 las cuales fueron conocidas como Cobol-Ansi.

Cobol significa "Lenguaje Común Orientado a los Negocios" por lo que se comprende que su misión en ser utilizado en la gestión y administración de los negocios, en contraste con otros lenguajes Cobol no fue concebido para calculos matemáticos ó científicos ya que de hecho solo dispone operadores para realizar los cálculos mas elementales, su empleo es mas apropiado para el proceso de datos en aplicaciones comerciales y la utilización de grandes Bases de datos.

Cobol es un lenguaje Multiplataforma y puede operar en Windows, Unix, MS-DOS, Linux, OS/400, S36, S34, VMS, Netware, Solaris etc., Actualmente Cobol puede ser usado en ambiente grafico y tambien en aplicaciones con Internet, integrar las aplicaciones de Cobol en el mundo Web es una realidad.

Algunas de las características del Lenguaje Cobol

- Cobol tiene futuro. La ANSI pondrá en marcha una versión actualizada de Cobol adecuada a los grandes adelantos de en tecnología.
- Cobol representa la experiencia. El 80% de los sistemas desarrollados en el mundo operan bajo este lenguaje.
- Cobol es sólido. Un gran numero de desarrolladores de software han desarrollado aplicaciones para ayudas en comprobación, análisis de aplicaciones, asistencia en la producción y reutilización de código.
- Cobol puede abarcar un amplio abanico de necesidades de procesamiento de datos. Las aplicaciones Cobol son ricas en posibilidades y funcionan tan bien al realizar las funciones vitales del negocio que los usuarios se resisten a prescindir de ellas.
- Cobol cumple su función de Soporte a aplicaciones empresariales.
- Cobol es fácil de aprender. Aún personas con pocos conocimientos técnicos han aprendido Cobol en muy poco tiempo, y se enseña en todo el mundo.
- Cobol es fácil de mantener. Los programas Cobol son fáciles de desarrollar y por su estructura de sintaxis es fácil de dar mantenimiento a sus aplicaciones.
- Cobol puede operar en múltiples plataformas.

Actualmente varias casas de software han desarrollado nuevas versiones de Cobol adecuándolo a las Metodologías actuales de la Programación Visual, entre las que podemos mencionar son:

- COBOL-WOW(RM/Cobol)
- ACUCOBOL-GT
- NET EXPRESS(MICROFOCUS)
- OBJECTIVE COBOL
- FUJITSU POWER-COBOL

Bill Gates dijo al publico en la Conferencia de Desarrolladores Profesionales reunida en Orlando Florida el pasado 12 de Julio del 2000 que Microsoft había escogido el Cobol de Fujitsu para apoyar sus planes en el desarrollo de la próxima Tecnología Microsoft.NET, "Hay todavía un nivel muy alto de uso de código Cobol, y por esto nosotros pensamos en el Cobol como una herramienta líder en el mercado".

Objective Cobol

Es un entorno visual de programación orientada a objetos, destinado al desarrollo rápido (RAD) de aplicaciones de uso general para Windows 95, 98 y NT. Con Objective Cobol se pueden desarrollar aplicaciones muy eficaces para Windows con un mínimo de código manual.

Objective Cobol proporciona una biblioteca de componentes reutilizables y una serie de herramientas de diseño RAD, entre ellas plantillas de aplicación y de fichas, además de Asistentes para la programación. Con estas herramientas y el compilador de código nativo de 32 bits se pueden crear prototipos de gran calidad, convirtiéndolos en aplicaciones robustas para satisfacer las necesidades de las empresas. Sus principales características son:

- Entorno de desarrollo integrado
- Diseño mediante el método de arrastrar y soltar
- Herramientas bidireccionales
- Compilador de código nativo
- Conectividad a las Bases de datos