

Ada, un lenguaje de aplicación militar, o una filosofía del Software

JOSE MANUEL DE LA RIVA GRANDAL, Tte. E.T.S. y
ANGEL GALVEZ MARTINEZ, Comandante Médico E.A.

(Ada es un lenguaje militar, iniciado por el Departamento de Defensa de USA, con el cual se espera recortar la escalada en el coste del software mediante la reducción del número de lenguajes de programación militar y dar respuesta a las necesidades de la Defensa, utilizando para ello métodos modernos de programación.)

(Era la tierra toda una sola lengua y de unas mismas palabras...) Génesis 11-12.

ANTECEDENTES HISTORICOS

En 1955 los costos del software eran un 10 por 100 menores que los del hardware, pero en 1975 eran ya tres veces mayores que los del hardware y se prevé que para este año sean nueve veces mayores (fig. 1).

A mediados de la década de los 70, el Departamento de Defensa de los Estados Unidos (DoD) gastaba al año 3 billones de dólares (10^9 \$) y habían inventariado alrededor del millar de lenguajes de programación en sus productos (fig. 2).

Para el año en curso el DoD tiene asignado un presupuesto de 200 billones de dólares, de los cuales un 10 por 100 están dedicados a Investigación y Desarrollo (I+D) y un 34 por 100 de éste a Informática y Electrónica.

También se detectó que mientras que la demanda de programas crecía a un ritmo del 12 por 100 anual, la productividad de los programadores lo hacía sólo en un 4 por 100 anual. Era pues necesario aumentar la eficacia de los programadores y/o automatizar la producción de software.

Con el fin de normalizar la situación, la Dirección de Investigación e Ingeniería de la Defensa de los Estados Unidos creó la comisión HOLWG (High Order Lenguaje Working Group), encargada de buscar un lenguaje de programación de Alto nivel (HOL), en Tiempo Real (fig. 3).

CITAS SOBRE Ada

1. Prof. C.A.R. Hoore. Programming Research Group. University Oxford. "Ada es el lenguaje de la tercera guerra mundial".

"Al principio estaba muy esperanzado con Ada, por su legibilidad, simplicidad, seguridad y definición formal de programas. Sin embargo, esos objetivos se están sacrificando en favor de la potencia y el peligroso manejo de excepciones".

2. Ichbiah J. Especificador del Ada.

"Nadie que haya tenido algo que decir sobre Ada puede decir que no se le haya escuchado".

3. Dr. T.A. Ward. Dpto. de Servicios Informáticos. University de North Iowa.

"Ada es un lenguaje sin final aparente, que se parece a un interminable contrato de Defensa".

4. Dr. E.W. Dijkstra. University de Texas.

"El PL/I fallaba porque no resolvía las necesidades del programador accidental y porque era insuficientemente transportable como para ser aceptado como un standard mundial; ahora estos problemas los ha resuelto Ada".

5. Massas E. Lott M. Ingenieros de Alsys.

"Ada es el lenguaje que permite la escritura de componentes lógicos a escala mundial".

6. Jordan A. University Carnegie-Mellon.

"Ada es un lenguaje que tendrá éxito tanto por el patrocinador del mismo como por los millones de dólares invertidos".

7. Verity J.W.

"Las nuevas ideas (sobre la ingeniería del software) que se encuentran serán más útiles para aquellos que estén impregnados de lenguajes más elegantes, como el Pascal e incluso el tan debatido Ada".

8. Ryer M. Director de Ada Systems Division for Intermetrics Inc.

"Ada al igual que el PL/I es un lenguaje complejo, pero se diferencia de éste en que las características están interrelacionadas".

9. Martín E.

"Ada tiene capacidad para poner orden en el caos que ha caracterizado hasta la fecha a las actividades de ingeniería del software militar del DoD".

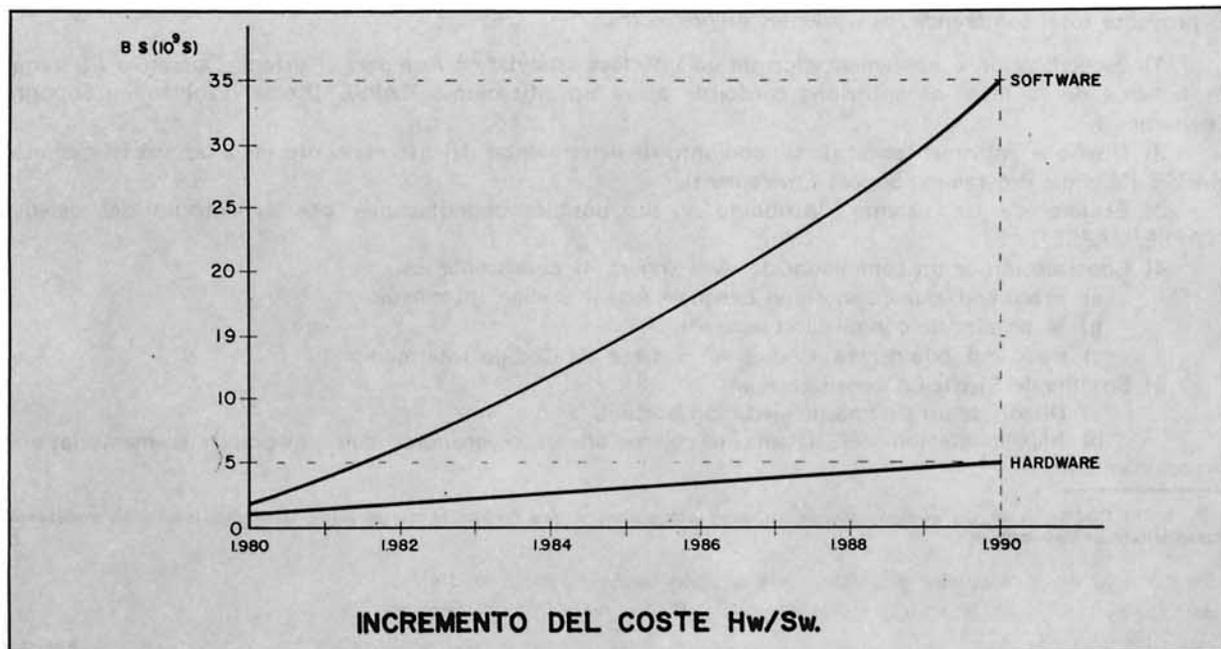
El HOLWG elaboró el informe TINMAN (1976); fijó los siguientes objetivos a cumplir:

1. Disminuir los costos del software.
2. Fiabilidad elevada.
3. Optimización del código generado.
4. Standarización del lenguaje.
5. Posibilidad de reutilización de los programas.
6. Transportabilidad a diferentes máquinas.
7. Legibilidad de los programas.

En 1977 aparecieron las especificaciones IRONMAN y en base a ellas se examinaron todos los lenguajes conocidos, seleccionándose el PL/I Algol 68 y Pascal. Este último fue escogido por la industria internacional como lenguaje base inicial. En febrero de 1978 fueron presentados los lenguajes desarrollados por las empresas previamente seleccionadas:

- | | |
|------------------|--------------------------|
| + Green Lenguaje | CII-HB/Jean Ichbich |
| + Red Lenguaje | Intermetrics/Ben Brosgol |

▼ Figura 1



+ Blue Lenguaje SofTech/John Goodenough
+ Yellow Lenguaje SRI International/Jay Spitzen

fueron evaluados de acuerdo con las especificaciones SANDMAN y STEELMAN, seleccionándose los dos primeros. Finalmente el 2 de mayo de 1979 se produjo la selección definitiva de acuerdo con las especificaciones PEBBLEMAN, recayendo el fallo en el equipo francés CII-HB.

(Dijéronse unos a otros "Vamos a edificarnos una ciudad y una torre, cuya cúspide toque a los cielos y nos haga famosos, por si tenemos que dividirnos por el haz de la tierra".) Génesis 11-12.

El lenguaje verde fue sustancialmente mejorado después de sufrir una evolución de 900 empresas de 15 países y finalmente fue bautizado con el nombre de Ada, y asignada la especificación militar estándar MIL-STD-1815, en honor de Lady Ada Augusta Lovelace (1815-1852), única hija legítima del poeta Lord Byron y reconocida hoy día como la pionera en el arte de la programación por sus estudios e interpretación de la Máquina Analítica de Charles Babbage, sacados a la vez por los ilustres matemáticos A. Turing y Von Neuman.

La síntesis del nuevo lenguaje fue publicado como "Preliminary Ada" y posteriormente se entró en un proceso de refinamiento siguiendo las especificaciones STONEMAN, el cual finalizó en julio de 1980. En julio de 1982 una revisión de MIL-STD-1815, fue publicada. El 17 de febrero de 1983 fue Standarizado por American National Standards Institute (ANSI/MIL-STD-1815A). Es un estándar de ISO (International Standard Organisation) y será un estándar de la NATO. El DoD planea una posible revisión del estándar a partir de 1988.

Ada Y EUROPA

La Defensa europea también ha dado los primeros pasos en la adopción de Ada como lenguaje de interés militar, siendo los alemanes occidentales los más drásticos sustituyendo su lenguaje militar Pearl por Ada, mientras que los británicos han adoptado Ada para sus sistemas de defensa junto al Coral. El lenguaje Ada será obligatorio en ambos países a partir de 1987.

La C.E.E. (en 1981) se planteó la necesidad de crear un lenguaje de programación europeo. Sin embargo, la idea no cristalizó y sólo en época reciente se abordó seriamente el problema a través del proyecto ACPD (Ada Compiler Development Proyet).

Dicho proyecto está siendo financiado por la C.E.E. en un 50% y el 50% restante por fundaciones europeas y las empresas Dansk Datamatic Center, Chistian Rousing AS e I.C. Olivetti, CS p A.

El objetivo del proyecto es la construcción de un entorno de programación con el propósito de facilitar el desarrollo y aplicación del Logical para sistemas que utilicen Ada y que se vayan a implantar en ordenadores con limitación de memoria y/o direccionamiento.

El proyecto total comprende los siguientes subproyectos:

1) Especificación e implementación de un interface estándar en Ada para el sistema Operativo y Sistema de ficheros de la máquina anfitriona conforme a las especificaciones KAPSE (Kernel Programing Support Enviroment).

2) Diseño e implementación de un conjunto de herramientas del sistema conforme a las especificaciones MAPSE (Minimal Programing Suport Enviroment).

3) Estudio de un sistema distribuido y sus posibles connotaciones con el entorno del sistema (KAPSE/MAPSE).

4) Construcción de un compilador de Ada (figura 4) consistente en:

- a) Front end, que convierte el Lenguaje Ada a Código Intermedio.
- b) Manejador de compilación separada.
- c) Back end, que genera Código-A* a partir de Código Intermedio.

5) Sistema de Ejecución consistente en:

- a) Diseño de un sistema de ejecución portátil.
- b) Implementación del sistema de ejecución en ordenadores con limitación de memoria y/o direccionamiento.

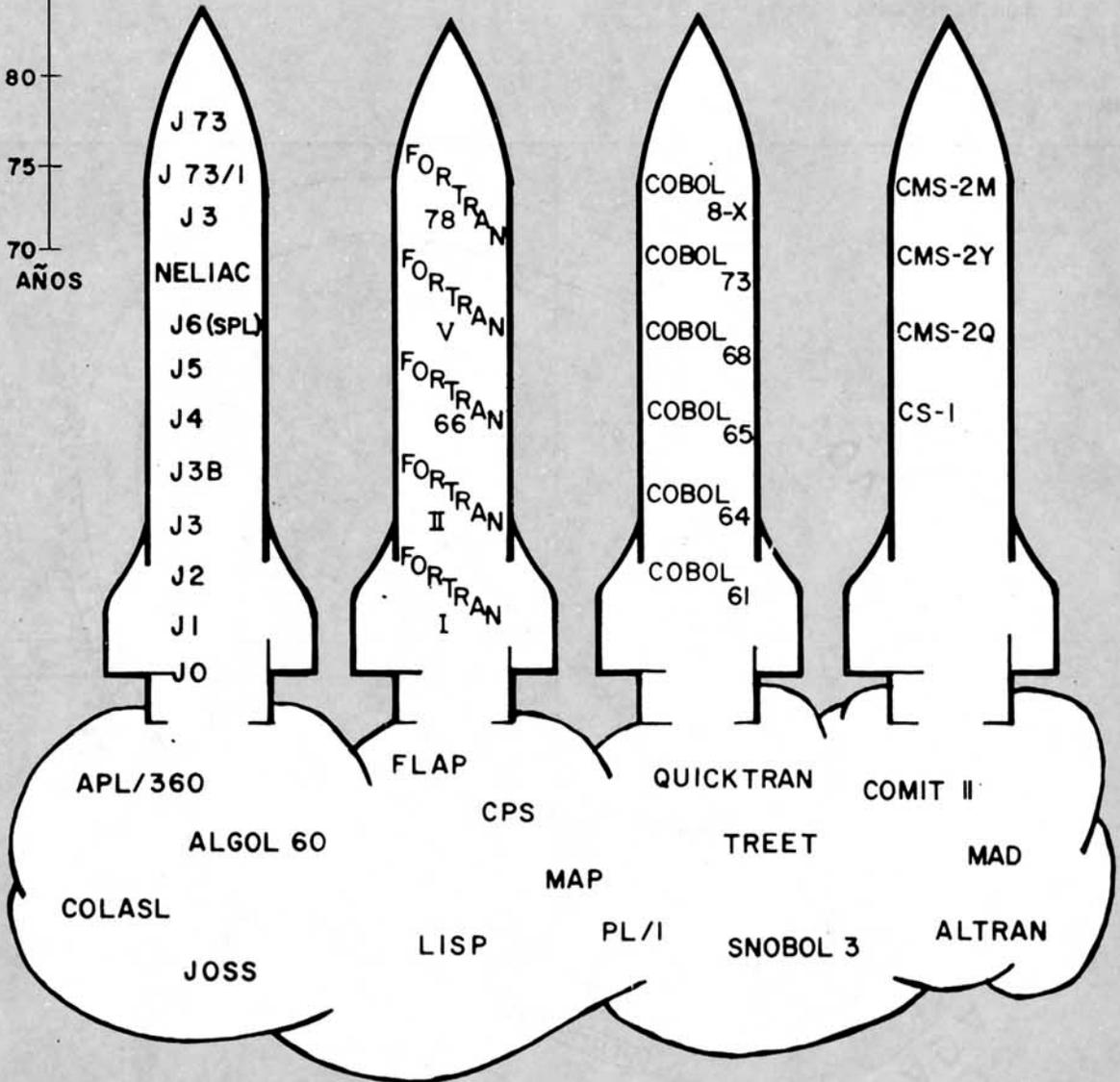
* El Código A es un código intermedio, muy compacto, y que facilita la definición e implementación de poderosas herramientas de depuración.

(Bajó Yahve a ver la ciudad y la torre que estaban haciendo los hijos de los hombres y se dijo: "He aquí un pueblo uno, pues tienen todos una lengua sola.") Génesis 11-12.

Figura 2 ►



90
75
70
AÑOS



TENDENCIA DE LOS LENGUAJES DE ALTO NIVEL EN EL DoD.

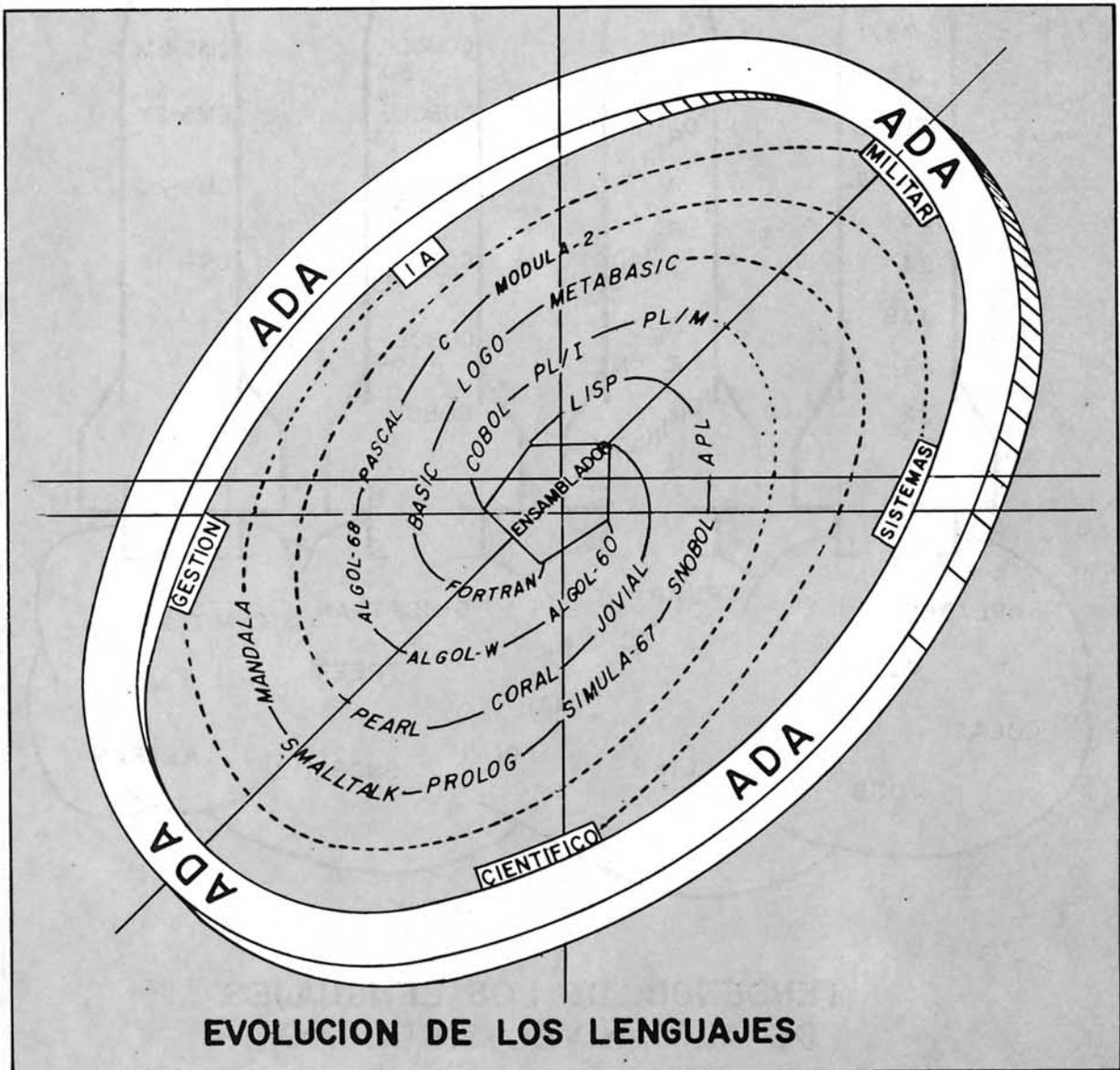
VENTAJAS E INCONVENIENTES DE Ada

Ada combina muchos de los conceptos de estructuras de programación que aparecieron inicialmente en los lenguajes estructurados como el Pascal, C, PL/I y HAL/S (usado en la lanzadera espacial). Entre dichas características destacan las siguientes:

Ventajas:

- 1) Modular.
- 2) Manejo de excepciones.
- 3) Compilación separada.
- 4) Módulos genéricos.
- 5) Portátil.
- 6) Depuración fácil.
- 7) Legibilidad alta.

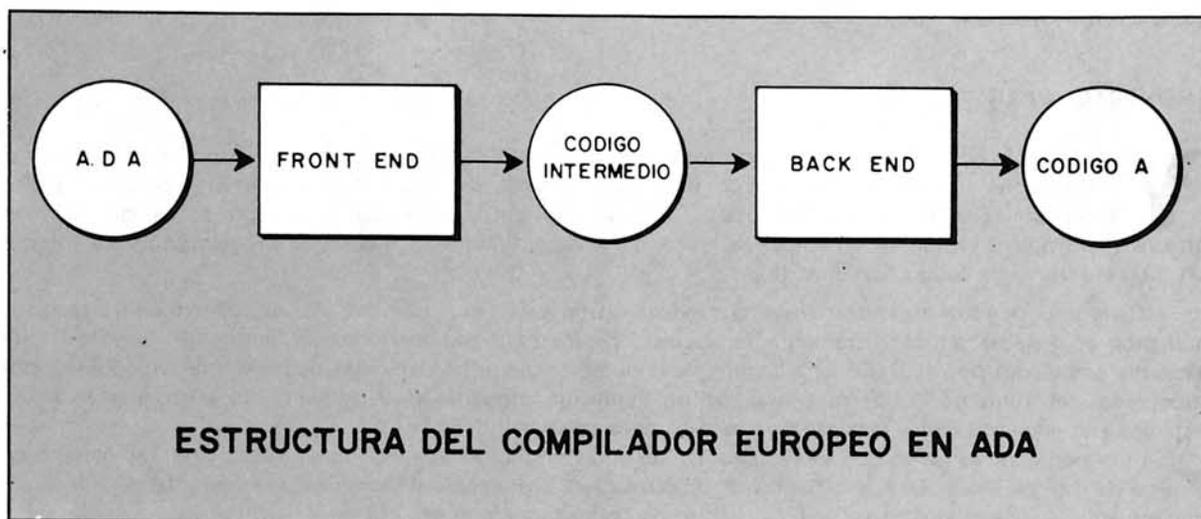
Figura 3 ▼



- 8) Genera Código óptimo.
- 9) Soporta Tipos abstractos de Datos.
- 10) Tiempo Real.
- 11) Mecanismo de concurrencia.
- 12) Interface con lenguaje de alto nivel (Pascal, C).
- 13) Interface directa con dispositivos físicos.
- 14) Metodología.
- 15) Separación física y lógica entre aplicaciones e implementación.
- 16) Orientado a sistemas de control (Embedded Systems).
- 17) Diseñado para "misiones críticas".
- 18) Expresivo.
- 19) Espacio de aplicación amplio.

Inconvenientes:

- 1) Complejo.
- 2) Algorítmico.
- 3) No ortogonal.
- 4) Tratamiento de E/S pobre.
- 5) Extenso.



▲ Figura 4

("Se han propuesto esto y nada les impedirá llevarlo a cabo.") Génesis. 11-12.

FUTURO DEL Ada

El DoD de los Estados Unidos ha realizado una inversión de 43.750 millones de ptas. para el desarrollo del Ada. Ello forma parte de un proyecto en el que ya se han invertido 27.475 millones de ptas. en el desarrollo de circuitos integrados de alta velocidad. Este proyecto incluye también el desarrollo de Sistemas Expertos e interface avanzados hombre/máquina, con una inversión total de unos 175 billones de ptas. (10^{12} ptas.). El DoD también ha fundado el Instituto Norteamericano para el desarrollo de la ingeniería del software dependiente de la Universidad Carnegie-Mellon, que indudablemente contribuirá al progreso de la tecnología de los soportes lógicos.

Igualmente las recomendaciones del informe británico ALVEY (octubre 82), sobre el desarrollo de la ingeniería del lógico para ordenadores de la V generación que incidirán sobre la Inteligencia Artificial, resaltan la utilización del Lenguaje Ada y del Sistema Operativo UNIX, aunque sin duda otros lenguajes como PROLOG, LISP y SMALTAX están en mejor situación en esa área.

Anna:

Uno de los lenguajes que ha nacido como extensión de Ada es el Anna (Annotated Ada) que es básicamente un lenguaje Ada con comentarios formales. Estos comentarios formales consisten en textos virtual Ada y anotaciones. Se definen la sintaxis y semántica de los diferentes tipos de anotaciones, como son las anotaciones declarativas, anotaciones de sentencia, anotaciones de excepción y anotaciones de visibilidad.

Anna incluye un pequeño número de atributos predefinidos, los cuales sólo pueden aparecer en las anotaciones.

La estructura léxica de Anna está diseñada de forma que las extensiones de Ada aparezcan con comentarios, siendo por tanto aceptables por cualquier compilador Ada.

La semántica de las anotaciones están definidas en términos de conceptos de Ada. Ello simplifica y facilita a los programadores Ada la especificación formal de sus programas mediante el uso del Anna.

Las herramientas de soporte que incluye el sistema Anna son:

. Analizador sintáctico, editor estructurado y detector manejador de errores y de inconsistencias entre especificaciones y código.

. Sistemas de chequeo en tiempo de ejecución, el cual traduce las especificaciones formales en rutinas de chequeo Ada.

. Sistemas de verificación formal, con interfaces con otras herramientas del APSE (Ada Programming Support Environment).

("Bajemos pues y confundamos su lengua de modo que no se entiendan unos a otros.") Génesis 11-12.

PROYECTO STARS

Ya hemos indicado que un entorno de programación APSE (Ada Programming Support Environment) proporciona un soporte automático para todo el ciclo de vida del software. El programa STARS (Software Technology for Adaptable Reliable Systems), ensancha este soporte, siendo sus tres objetivos principales: aumentar los recursos humanos, aumentar las herramientas de programación y alentar el uso más extensivo de dichas herramientas.

Cada uno de estos tres objetivos está subdividido en varios subobjetivos. Así el objetivo de los recursos humanos se desglosa en un incremento en el nivel de los expertos y en una expansión de la base de los expertos empleados por el DoD. El aumento de la potencia de las herramientas de programación conlleva tres subobjetivos, el aumento de herramientas de programación específicas, el aumento de la programación de sistemas y el aumento en las técnicas de dirección de proyectos.

Finalmente, el aumento de uso de tales herramientas implica el alcanzar metas tales como incremento en el nivel de integración de las herramientas de programación, incremento en los niveles de automatización de las herramientas e incremento en la facilidad de uso de las herramientas. ■

BIBLIOGRAFIA

Libros:

- 1) Barnes, J.G.P. Programing in Ada, 2-1984.
- 2) Cherry, G.W. Parallel programing in ANSI standard Ada, 1984.
- 3) Downes y Golsack. Programing embedded systems with Ada, 1982.
- 4) David, D.J. Le Lenguaje Ada, 1983.
- 5) Gehani, N. Ada. An advance introduction including. Reference Manual for the Ada Programing lenguaje, 2-Ed.
- 6) Hibbard, P. Studies in Ada style, 1983.
- 7) Le Beux, P. Introduction a Ada, 1982.
- 8) Le Verrand, Le lenguaje Ada manual d'evaluation, 1982.
- 9) Medermind y Ripken. Life cycle support in the Ada environment, 1984.
- 10) Nissen y Wallis. Portability and style in Ada, 1984.
- 11) Olsen y Whitehill. Ada for programers, 1983.
- 12) Pylar, I.C. The Ada programing language.
- 13) Ledgard II. Introducció al Ada, 1983.

Manuales:

- 1) Reference manual for the Ada programing language. ANSI, 1983.

Revistas específicas:

- 1) Ada letters, ACM.
- 2) AdaData. International Resource Development Inc. Norwalk, Ct.
- 3) Proceeding Ada in use. International Conference, París, 1985.
- 4) Ada Information Clearinghouse Newsletter, Ada Joint Program Office.

Artículos:

- 1) Sankar. S. et al. Implementation of Anna. Proceeding of the Ada. International Conference, París, 1985.
- 2) DDC Ada compiler systems separate compilation handler, Functional specification. Dansk Datamatik Center, 1984.
- 3) Computer, Vol. 16, No. 11, november 1983, pp. 9-104.

Organizaciones profesionales:

- 1) Ada JUG (Ada-Jovial User's Group).
- 2) SigAda.
- 3) AdaTec.