

# Compiladores Ada: estado actual

JOSE MANUEL DE LA RIVA GRANDAL, Tte. E.T.S.; FERNANDO ENRIQUEZ RODRIGUEZ,  
Cte. Médico E.A. y MIGUEL ANGEL GOMEZ MARINO, Cap. Médico E.A.

## CONCEPTOS BASICOS SOBRE COMPILADORES

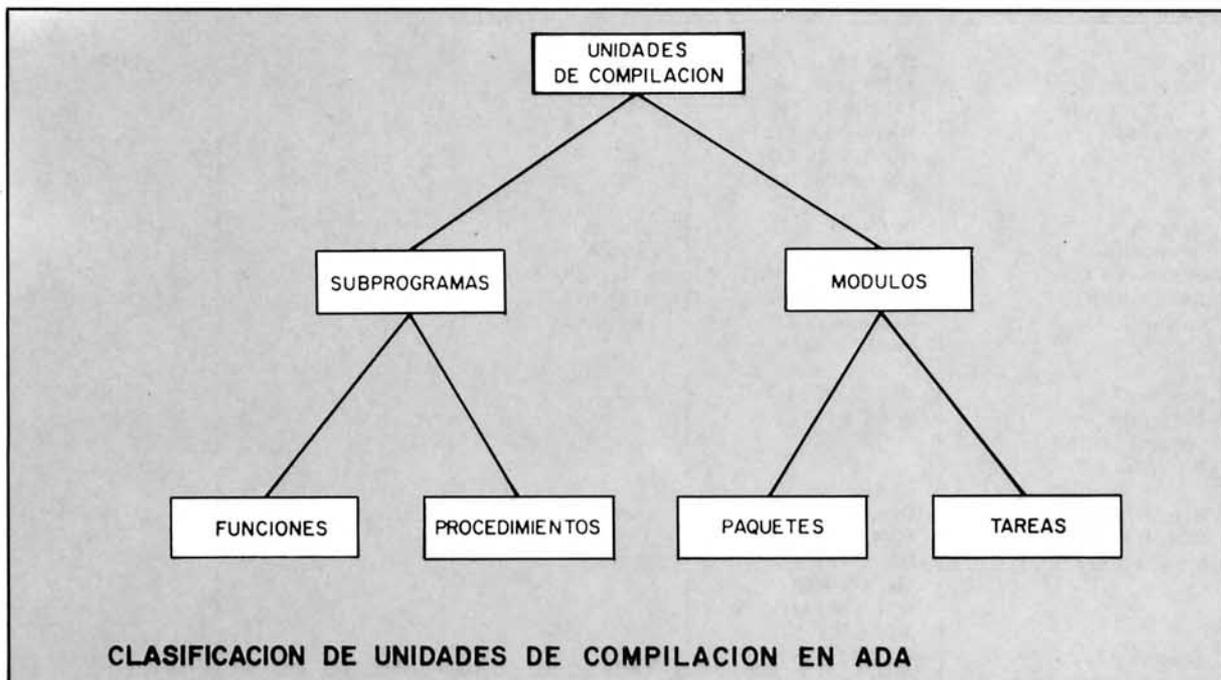
**E**l primer compilador moderno se lo debemos a J. Halcombe Laning, Jr. y Neal Zierler del Laboratorio de Instrumentación del MIT (Masachuser Institute Tecnology), los cuales en el año 1952 desarrollaron un compilador algebraico "George", para el lenguaje MAC (Mathematical Automatic Coding), que estaba adaptado particularmente al manejo de vectores tridimensionales. El George fue utilizado en Whirlwind (Prototipo de Defensa Aérea) y ha estado en uso hasta fecha muy reciente.

Desde entonces hasta hoy la evolución de los lenguajes y sus programas traductores ha sido constante. En la actualidad los compiladores son unos programas que traducen programas escritos en un lenguaje de alto nivel (lenguaje fuente) a lenguaje objeto, el cual puede ser un código intermedio o el código máquina. El lenguaje en el que está escrito el compilador se denomina lenguaje de Implementación, y puede ser el mismo que el fuente.

## APORTACIONES DE LOS COMPILADORES Ada

**A**unque el lenguaje Ada está en desarrollo desde 1975, sólo hasta fechas muy recientes se ha dispuesto de compiladores operacionales completos.

A finales de 1984 la construcción de compiladores Ada llevaba un cierto retraso, pues sólo estaban validados por el Departamento de Defensa cuatro compiladores en vez de los diez inicialmente previstos. Las



▲ Figura 1

razones que esgrimen los diseñadores para justificar este retraso es esencialmente el de que Ada es un lenguaje complejo que incorpora muchas características de las que los lenguajes anteriores carecían.

Hoy en día existen un conjunto numeroso de compiladores validados, no sólo en los EE.UU., sino también en Francia, República Federal de Alemania y Dinamarca (ver tabla 1).

Para hacernos una idea de la envergadura del proyecto de un compilador, mientras que un traductor de Pascal tiene por término medio 5.000 líneas, un compilador Ada está en torno a las 100.000 líneas. Además, a juicio de los expertos estos compiladores no eran de buena calidad, por ser lentos y producir un código deficientemente optimizado, condición intolerable en los sistemas empotrados de aplicación militar.

Entre las novedades que aporta el compilador Ada, está la compilación modular y la condicional.

Una unidad de compilación es una unidad de estructura para desarrollar un programa y de compilación separada. Puede constar de un subprograma o de un módulo, a su vez el subprograma es o una función o un procedimiento; y el módulo, un paquete o una tarea (fig. 1). Un paquete se usa para especificar una colección de fuentes lógicamente relacionadas. Una tarea es una unidad de programa modular que se usa en la especificación de concurrencia.

El compilador Ada incluye un manejador de compilación separada, que es una característica estándar de Ada. Este manejador controla el almacenamiento y/o la recuperación de toda la información referente a los módulos compilados, incluyendo los descriptores de los módulos con el fin de poder verificar el correcto interface entre los mismos.

Un compilador condicional es aquel que previo chequeo a una expresión condicional genera o no el código correspondiente a ese estamento o subprograma. Con este recurso se obtiene una optimización del código generado por el compilador, puesto que cuando la condición no se cumple el código objeto se omite.

TABLA 1  
COMPILADORES Ada VALIDADOS

Firma/Compilador	Anfitrión	Destino	Versión Test	Fecha
Alsys (Francia) Alsy COMP-0001 Versión 1,0	VAX 11/785, 11/780 11/750, 11/730 Micro VAX I (VMS 4,0)	ALTOS ACS 68000 (ALTOS Versión 1)	1,5	08/12/84
Dansk Datamatik Center DDC Ada compiler VAX 11, Realización 1.1	VAX 11/750 (VMS 3,5)	igual	1,4	19/10/84
Digital Equipment Co, DEC VAX-Ada Compilador	VAX 11/785 11/782, 11/780, 11/750, 11/730, Micro VAX I (VAX VMS 4.0) y (Micro VMS 1.0)	igual	1,4	19/09/84
Honeywell Information Systems GCOS6 Ada Compiler versión 1.1	DPS 6/95, 6/75 6/45 y microSystem 6/20 (GCOS6 MOD400, Realización 3.0 Puesta al día 4)	igual, plus microSystem 6/10 (GCOS6) MOD 400, Realización 3.0)	1,4	14/12/84
New York University Courant Institute NYU/Ada ED 1.4	DEC VAX 11/780 (VAX VMS 3.5)	igual	1,4	06/08/84
ROLM/Data General ROLM/Ada Compiler Versión 5/234	Data General MV4000, 6000 8000, 8000-II, 10000 y ROLM MSE-800 (ADS/VS- Ada 3.12)	igual	1,4	21/05/84
Telesoft Ada Compiler, Versión 2.0a6 (UNIX)	VAX 11/780 (UNIX 4.2 bsd)	igual	1,4	05/02/85

## VALIDACION DE LOS COMPILADORES Ada

Una vez desarrollado el compilador Ada, la siguiente fase es la validación. Esta etapa tiene por misión el detectar todos los posibles errores que tenga el compilador tanto en tiempo de compilación como de ejecución.

Está comúnmente aceptado que existe un error en un compilador.

- 1) Para una entrada correcta, la salida es incorrecta.
- 2) Para una entrada incorrecta, la salida es correcta.

La técnica seguida hasta ahora para validar un compilador es someter al mismo a un conjunto de programas de pruebas o baterías de test suficientemente amplios para mostrar todos los posibles errores que pueda contener el traductor. Sin embargo, es conocido que no es económicamente rentable ni informáticamente posible el elaborar programas de pruebas que capturan todos los errores; por ello la estrategia empleada hoy en día en el diseño de compiladores es crear test que detecten los errores cuya probabilidad de aparecer en explotación sean altos y que el resto sean validados por el propio usuario y posteriormente corregido por la casa que elabora y/o comercializa el producto.

La forma tradicional de validación de cualquier producto de software sigue la línea clásica de hacerlo mediante baterías de test, con distintos juegos de datos de tipos diversos que simulen la realidad.

Esta técnica ha resultado pobre e insatisfactoria por la baja fiabilidad, pues servía para detectar la presencia de errores, pero no para asegurar su ausencia.

La necesidad de conseguir programas fiables, dio lugar a dos métodos de validación:

- a) Métodos de programación estructurada.
- b) Métodos formales de procedimientos.

Firma/Compilador	Anfitrión	Destino	Versión Test	Fecha
TeleSoft Ada Compiler, Versión 2.0a6 (VMS)	VAX 11/780 (VMS 3.4)	igual	1,4	05/02/85
TeleSoft Ada Compiler, Versión 2.0a6 (SUN Microsystems)	SUN 120 Motorola M68010 (1.2 bsb UNIX-SUN Versión 1.1)	igual	1,3	10/08/84
TeleSoft Ada Compiler, Versión 2.0a1 (Callan)	Callan Unistar 300 (UNIX Systems V, Versión 1.1a)	igual	1,3	10/08/84
TeleSoft Ada Compiler, Versión 2.0a1 (LabTEK)	LabTEK Motorola 68000 (ROS)	igual	1,3	21/05/84
TeleSoft Ada Compiler, Versión 2.0a1 (O-Bus)	O-Bus Motorola 68000 (ROS)	igual	1,3	21/05/84
University of Karlsruhe (Alemania) Siemens BS2000 Versión B40404	Siemens 7.571 (BS2000 7.1)	igual	1,4	14/11/84
University of Karlsruhe (Alemania) VAX 11, Versión V1.0	VAX 11/750 (VMS 3.0)	igual	1,4	12/11/84
U.S. army CECOM/SofTech ALS AdaVAX, Versión 1.1	VAX 11/780 (VAX VMS 3.6)	igual	1,4	31/12/84
Verdix Ada Compiler VAda-010-0101, Versión V03,04	VAX 11/750 (UNIX 4.2 bsb)	igual	1,5	15/03/85
Nota: Sistemas Operativos entre paréntesis.			Fuente: Ada Joint Program Office (11/04/85)	

Los métodos de verificación del compilador Ada siguen la filosofía tradicional, mediante baterías de programas de pruebas y fueron encomendados por el DoD a la empresa SofTech. Su misión consistió en confeccionar el paquete de test del traductor de Ada, consistente en un juego de 1.850 programas de pruebas, clasificados en seis tipos de test, de acuerdo con la naturaleza de su comportamiento (Tipos A, B, L, C, D, E), de los cuales el 80% son aplicables a todos los compiladores Ada, y el resto de los test se exigen dependiendo de las características particulares que el implementador haya escogido.

Es interesante reseñar que en el procedimiento oficial de validación impuesto por el Pentágono, no incorpora requerimientos de velocidad, hecho que incide negativamente en el desarrollo de programas para misiones críticas, en los que la velocidad es muy importante.

La validación no es un concepto peculiar de Ada, pues ya anteriormente el DoD lo utilizó en el desarrollo del COBOL. Para el compilador Ada el Departamento de Defensa de EE.UU. ha creado un organismo AJPO (Ada Joint Program Office) para velar por la estandarización del lenguaje Ada. Su misión es realizar la prueba oficial de validación de los distintos compiladores desarrollados por las empresas de software. Solamente expide el Certificado Oficial de Validación (fig. 2) si el compilador a examen pasa todos y cada uno de los test sin el menor error. El certificado tiene únicamente validez anual.

## PANORAMICA DE COMPILADORES COMERCIALES Ada.

La tabla 2 resume una selección de las experiencias desarrolladas en Ada. (No se incluye los 5 millones de dólares gastados en generar las especificaciones del lenguaje.)

### \* Compilador Rolm:

El compilador Ada, desarrollado por la casa Rolm, está orientado a correr en superminis de 32 bits equivalentes a los de Data General.

Solamente en el diseño del compilador se han gastado aproximadamente 20 años/hombre sin contar el tiempo gastado en el desarrollo del soporte. El compilador fue anucado en octubre de 1982 y validado en junio de 1983. Fue el primer compilador validado, aunque ya antes lo fue el intérprete Ada de NYU Ada/ED.

El computador que utiliza es el MV 10000 de Data General de velocidad de compilación de 120 líneas por minuto de CPU, mientras que el ordenador VAX 11/780 DEC es, aproximadamente, la mitad de rápido y su velocidad de compilación es 60 líneas por minuto de CPU. Su lenguaje intermedio es el DIANA (Descriptive Intermediate Attributed Notation for Ada).

### \* Compilador Gensoft

Este compilador es una generación de código verdadera que traduce un código intermedio similar al DIANA a código Pascal-P y el hardware ejecuta directamente sin un intérprete. La máquina procesa a 1/8 de la velocidad del VAX 11/780 DEC. Como lenguaje de implementación usa el Pascal, tal como Irvine y a diferencia del resto de las compañías que usan Ada. El sistema y entorno del lenguaje les ha llevado 8 años/hombre, de los cuales han pasado dos.

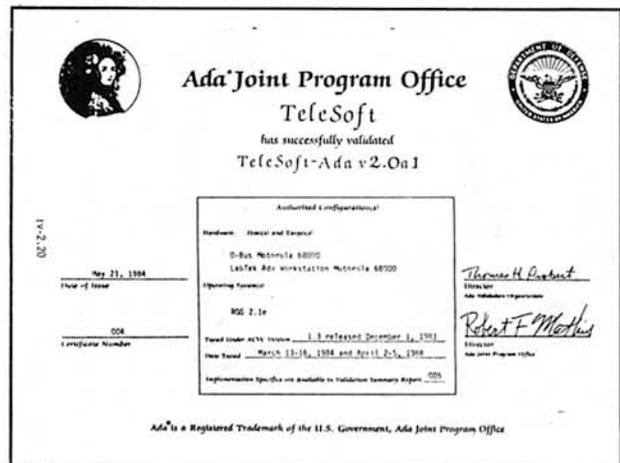
El compilador no está optimizado, pero es eficiente y ofrece un número de características asociadas con la optimización. La velocidad de compilación es de 120 líneas de programa fuente por minuto.

### \* Compilador Telesoft

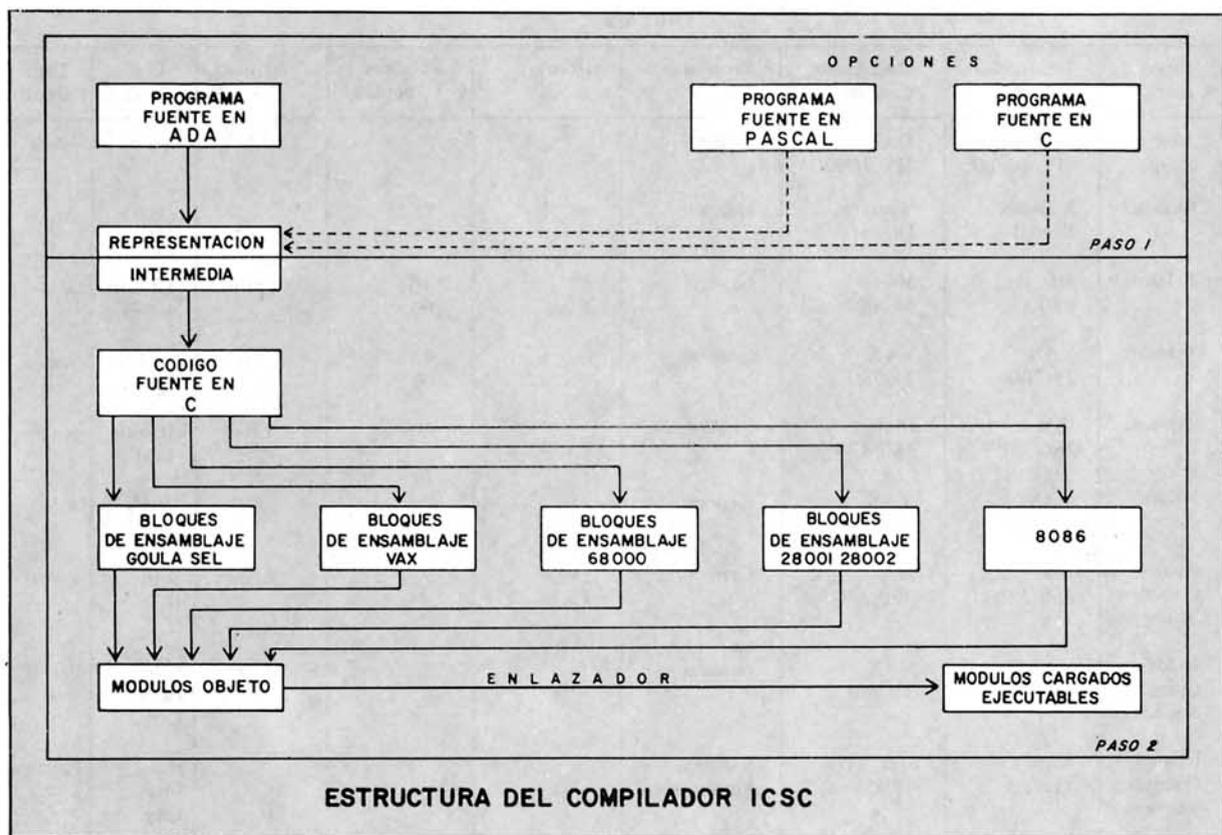
El compilador Telesoft usa dos lenguajes intermedios: Un Front End traduce código fuente, primero al lenguaje intermedio Hi-Form, y de éste al Low-Form.

El compilador está escrito en Ada, habiéndose utilizado en su construcción la técnica del bootstrapping. Primeramente se desarrolló un subconjunto de Ada en Pascal. El esfuerzo se tasó en 20 años/hombre, sin contar el diseño del compilador de un subconjunto de Ada.

Los compiladores para familia Motorola 68000 corren bajo sistema operativo ROS de Telesoft y fueron validados en mayo 1984 y bajo sistema operativo Unix en agosto 1984. Los compiladores para VAX DEC corren bajo Unix y VMS y fueron validados en enero de 1985. Los compiladores para la familia IBM 370 corren bajo MVS y CMS, validándose en mayo 1985.



▲ Figura 2



▲ Figura 3

**\* Compilador Softech**

Desde junio de 1980 esta empresa trabaja en el desarrollo de Ada para VAX 11/780. Su código intermedio es DIANA y su velocidad de compilación es de 400 líneas/min. Tiene incorporadas técnicas de optimización global y locales. Fue validado en diciembre de 1984.

**\* Compilador Intermetrics**

En octubre de 1980 esta empresa empezó con el diseño de un compilador Ada para IBM 370, la implementación llegó en abril 1982 y la validación se obtuvo en septiembre 1985. El compilador correrá bajo los sistemas operativos MVS de IBM y UTS de Amdahl.

Tal como Telesoft, usa dos lenguajes intermedios. El Front End del compilador traduce el código fuente, primero a DIANA y éste a BILL. El lenguaje de implementación del compilador es Ada, y su construcción se basa en la técnica de bootstrapping.

**\* Compiladores ICSC**

Irvine Computer Sciences Corporation (ICSC) ha desarrollado compiladores Ada para un gran número de máquinas.

El Front End del compilador traduce código fuente a código intermedio ICSC, el cual es independiente de la máquina destino. Después traduce el código intermedio ICSC a código C. Este código fuente C es un formato de instrucciones básicas para computadores que corren sobre el sistema operativo Unix, es un lenguaje de máquina universal que permite hacer portátil el compilador en sistemas futuros. Compiladores de IBM y Gould tienen traductores que producen C como segundo lenguaje intermedio.

Una diferencia de la organización y estructura de estos compiladores es que son de dos pasadas en vez de seis o siete que es lo común, con lo cual su velocidad es mayor aunque más complejo. El compilador está dividido en dos programas: El primero es un Front End y el segundo un Generador de código. En la primera pasada se ejecutan todas las tareas analíticas que son características de los lenguajes de alto nivel, siendo su salida una representación intermedia, igual para cualquier máquina.

En la segunda pasada esta representación intermedia se traduce a código fuente para un computador particular, y el bloque de ensamblaje para esa máquina particular compila y enlaza en código C en módulos ejecutables (fig. 3).

TABLA 2

Firma comer.	Ordenador anfitrión	Ordenador destino	Propósito	Hombre/año	% Palabras % velocidad	Líneas/ min.	Len inter.	Len. impl.
Rolm Corp.	Data Gen MV/10000	D.G. MV/10000	General		? ?	1200	DIANA	Ada
Gensoft	Western Digital	Western Digital	General	8	? ?		como-DIANA	Pascal
Telesoft	MC 68000	MC 68000	General	20 Total	40% 100%	500	Hi-For. Lo-For.	Ada
Telesoft	VAX 11/780	VAX 11/780	General		40% 100%	600	Hi-For Lo-For.	Ada
Telesoft	IBM 360/370	IBM 360/370	General		40%	1500	Hi-For. Lo-For.	Ada
Softech	VAX 11/780	VAX 11/780	General			400	DIANA	Ada
Irvine Computer Sciences	IBM 360/370	IBM 360/370	General	25 Total		1300	ICSC IL C	Pascal
Irvine Computer Sciences	VAX 11/780	VAX 11/780	General			1500	ICSC Inter Larg.	Pascal
Irvine Computer Sciences	VAX 11/780	MIL-STD-1750A	Sistemas Empotrados			?	ICSC Inter. Larg.	Pascal
Irvine Computer Sciences	MC 68000	MC 68000	General			800	ICSC Inter. Larg.	Pascal
Irvine Computer Sciences	Zilog Z8000	Zilog Z8000	General			1500	ICSC Inter. Larg.	Pascal
Irvine Computer Sciences	Gould Sel	Gould Sel	General			1500	ICSC IL C	Pascal
Intermetrics	IBM 360/370	IBM 360/370	General		20% 20%	2000	DIANA BILL	Ada

Es decir, el primer paso es el mismo para todas las máquinas, el segundo paso traduce la salida del primer paso en código para cada máquina en particular con el relativo alivio. Los generadores de código se desarrollan en unos cinco meses.

Aunque Ada es tan rápido como cualquier lenguaje de alto nivel, la velocidad en una aplicación particular depende de la complejidad del programa, de las características del ordenador y del tamaño de la memoria. En el caso de estos compiladores su velocidad es muy elevada de aproximadamente 1.500 líneas de código máquina por minuto.

Estos compiladores están escritos en Local networking System-Pascal que es una variación del lenguaje Pascal.

Existen compiladores de Ada para los siguientes computadores:

Gould Inc. SEL, VAX/11 DEC, Motorola 68000 de 16 bits Z8000 de Zilog y el procesador 8086 de Intel.

**BIBLIOGRAFIA:**

- 1) Berard, E.V.: "Ada Steps Out." Datamation, Vol. 29, No. 9, Sep. 1983, pp. 114-123.
- 2) Brender, R. F. e I.R. Nassi: "What is Ada", Computer, Vol. 14, No. 6, June 1981, pp. 17-24.
- 3) Computer, Vol. 14, No. 6, June 1981, pp. 11-64.
- 4) SIGPLAN Notices, "Preliminary Ada reference manual". Vol. 4, No. 6, 1979.
- 5) Stenning. V. et. al. "The Ada environment: A perspective". Computer, Vol. 14, No. 6, June 1981, pp. 26-36.