

LENGUAJES DE PROGRAMACION CON ORDENADORES

Por JOSE LUIS SALILLAS

Ayudante de Obras Públicas
Gabinete de Cálculo del C. E. E. de O. P.

En la última década el cálculo por medio de ordenadores electrónicos ha adquirido una enorme difusión, tanto en sus aplicaciones técnicas como en procesos administrativos. Sin embargo, su uso ha estado frenado por los complicados lenguajes que entienden estos cerebros. Para remediarlo se han creado lenguajes simbólicos cada vez más elaborados, que facilitan la programación. Un lenguaje capaz de describir cualquier tema es necesariamente complicado y no suficientemente potente para tratar problemas particulares, por lo que es conveniente disponer además de otros especializados. Como un ordenador no entiende directamente estos lenguajes simbólicos, se escribe para cada uno un programa traductor. El desarrollo de los lenguajes simbólicos multiplica la necesidad de traductores. Un lenguaje para escribir programas traductores puede ser la mejor solución. En este artículo se describe la utilización de éste para solucionar los problemas apuntados.

Lenguajes O.

Los lenguajes artificiales, también llamados no naturales, son los construidos por el hombre de una forma sistemática y con un propósito determinado. Son lo opuesto a los lenguajes naturales (español, inglés, francés...) construidos por el hombre de una forma no sistemática y en un deseo de comunicación. Lenguajes como el esperanto o el *basic english* son artificiales, tanto por haber sido construidos sistemáticamente como por tener el definido propósito de llegar a ser lenguajes universales; pero se distinguen de los lenguajes que vamos a tratar en que sus temas son de carácter general.

Los lenguajes que comentaremos (lenguajes C) son lenguajes completos de tema especializado. Decimos que son lenguajes completos porque, a diferencia de una jerga, todos los elementos le pertenecen. Nótese que el lenguaje utilizado por abogados, médicos, matemáticos, no es entendido por individuos no profesionales aunque hablen el mismo idioma, y, en efecto, estos lenguajes pertenecen a una misma lengua.

Nuestros lenguajes tratan de un tema específico como los lenguajes de abogados, médicos, etc. Y de estos lenguajes estamos principalmente interesados en aquéllos que pueden ser interpretados por ordenadores (FORTRAN, COBOL, MAD...).

También la matemática es un lenguaje y podría pertenecer a los lenguajes C por ser un

tema muy especializado; sin embargo, como lenguaje en sí no ha sido sistemáticamente construido y en su uso normal tampoco es completo (obsérvese cómo se enuncia un teorema o se razona un resultado).

Sin embargo, la matemática es un buen ejemplo de cómo usamos un lenguaje no sólo como medio de comunicación, sino como poderoso instrumento de razonamiento. Algunos de los lenguajes C son principalmente poderosos instrumentos de cálculo (FORTRAN), otros transmiten o sistematizan la información (I P L).

La transformación de una información es un proceso. Este proceso puede consistir en cálculos aritméticos, decisiones lógicas, traducción de signos, etc.

Para enunciar estos procesos utilizamos lenguajes más o menos elaborados, a veces simples dibujos; otros, gráficos con declaraciones verbales, etc. (hoy se habla del lenguaje cinematográfico).

La utilidad de un lenguaje estriba en la potencia de sus símbolos (piénsese en el '0'), en sus reglas de transformación (lógica simbólica) y en la significación de sus objetos (conceptos).

Se ejecutan los procesos por procedimientos manuales o mecánicos. La utilización de mecanismos multiplica la eficacia y corrección de los procesos. La automatización de los procesos por medios mecánicos (incluidos electrónicos y otros) es el sumum.

Los procesos enunciados en lenguajes C y ejecutables en ordenadores constituyen uno

de los grupos de procesos automáticos más eficientes. A los lenguajes C ejecutables en ordenadores los llamamos lenguajes O.

El tema de los lenguajes O puede ser bastante general: resolución de cálculos matemáticos (ALGOL, FORTRAN...), resolución de procesos de tipo comercial (COBOL...), análisis de símbolos y traducciones de idiomas (COMIT...), etcétera.

Por ser lenguajes C, a estos lenguajes en la terminología anglosajona se les llama de orientación al problema. Pero hay lenguajes orientados de una forma mucho más especializada a los problemas (COGO, STRESS...).

Estos son lenguajes que permiten resolver problemas de un campo muy especializado. Así, el COGO resuelve problemas de geometría analítica que ocurren principalmente en trabajos topográficos, el STRESS permite calcular estructuras de una gran variedad, el FORMAC permite calcular derivadas y otras transformaciones analíticas. Hay cientos de estos lenguajes y constituye el medio mejor de aplicar el cálculo electrónico, el disponer un lenguaje especializado para tratar un determinado tipo de problemas. Así, unos grandes almacenes, con su gestión dirigida por un ordenador electrónico, deberá disponer de un lenguaje que interprete las cuestiones significativas de este tipo de empresa. Lo mismo para tratar problemas de estrategia militar, de administración de fincas, etcétera. No tardará el día en que un contable especializado en empresas (por ejemplo) precise además del bagaje de conocimientos habitual en este campo, el lenguaje O que le permita tratar y resolver sus problemas.

Lenguaje MOVTER.

Como ejemplo que aclare estas ideas veremos cómo un ingeniero enunciaría el cálculo del coste del movimiento de tierras en un lenguaje O, que vamos a llamar MOVTER. Transcribimos sólo un fragmento del posible programa:

```
DESDE KM. 200,000 a KM. 202,800.  
CANAL SECCION CIRCULAR TIPO 3, RADIO 4,5.  
BANQUETA DERECHA TIPO NORMAL.  
CUNETAS BANQUETA TIPO 1.  
CUNETAS CORONACION TIPO 2.
```

```
MURO TIPO ANGUITA SI PROF DESM >  
4 y < 8.  
MURO TIPO REGLADO SI PROF DESM > 8.  
TALUD DESM 3/1.  
TALUD TERR 1/1.  
ESCALAS H = 1/100.  
EJECUTE.  
PERFILES.  
VOLUMENES.  
COSTES.  
FINAL.
```

El resultado de este programa sería el dibujo de los perfiles transversales por medio de un *plotter*, el listado de superficies y volúmenes parciales y acumulados del movimiento de tierras y, por último, un listado del detalle de costes del movimiento de tierras y las obras de fábrica.

En el MOVTER existen palabras obligatorias y otras de uso voluntario. Son obligatorias entre las utilizadas en el fragmento anterior: DESDE, CANAL, TIPO, RADIO, BANQUETA, DERECHA, Y, CORONACION, CUNETAS, MURO, SI, PROF, DESM, TALUD, TERR, ESCALA (S), H, V, EJECUTE, PERFILES, VOLUMENES, COSTES, FINAL.

Voluntarias: KM, A, SECCION, CIRCULAR. Los signos ' / ', ' = ', ' . ' son obligatorios y el uso de la coma es voluntario.

También se utilizan variables. NORMAL, ANGUITA, REGLADO, son variables definidas en otra parte del programa por una lectura de datos o por una declaración de asignación.

El programa anterior sería sólo el enunciado del proceso, su ejecución (ver fig. 8) se efectuaría en un ordenador o, más exactamente dicho, en un equipo de proceso de datos, puesto que además de la unidad de cálculo con su correspondiente memoria precisará de un *plotter*, donde se dibujarán los perfiles; una impresora, donde se listarán los resultados, y una lectora donde se leerán las fichas de datos. Además, fuera de línea (*off-line*), dispondrá de perforadoras, verificadoras y clasificadoras, donde se preparan los datos.

Ensambladores y compiladores.

Sin embargo, ningún ordenador entenderá el lenguaje MOVTER. En realidad, un ordenador entiende un solo lenguaje que le es propio. A estos lenguajes, que son lenguajes O, se les

llama lenguaje máquina (lenguajes M) y se les representa por medio de unas configuraciones de unos y ceros (*bits*) que significan instrucciones, direcciones, números, signos y letras.

Como la programación en estos lenguajes es penosísima, no se utilizan en la programación ordinaria, sino que para utilizar toda la capacidad de la máquina se usan unos lenguajes A llamados ASSEMBLER, S P S etc., que utilizando palabras, signos aritméticos y sintácticos permiten programar de una forma análoga, pero mucho más fácil que en un lenguaje M.

Por lo antes dicho, un lenguaje A no es entendido directamente por un ordenador, pero con el lenguaje M se construye un programa (fig. 1) que traduce un programa escrito en len-

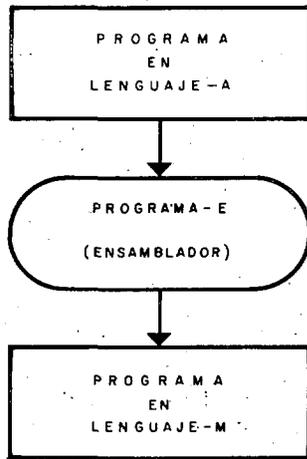


Figura 1.

guaje A a otro en lenguaje M. La característica principal de estos programas, llamados ensambladores, es que traducen cada instrucción del lenguaje A a una sola del lenguaje M. A este proceso se le llama ensamblar.

Entonces, si un programa (ensamblador) permite traducir un programa escrito en lenguaje A a otro en lenguaje M, otro podría permitir traducir un programa escrito en MOVTER a otro en lenguaje M. A estos programas (programas C) se les llama compiladores, y la característica principal que les distingue de los ensambladores es que traducen cada unidad lingüística del lenguaje traducido en una o más instrucciones en lenguaje de máquina.

El proceso podría haberse realizado también como indica la figura 2.

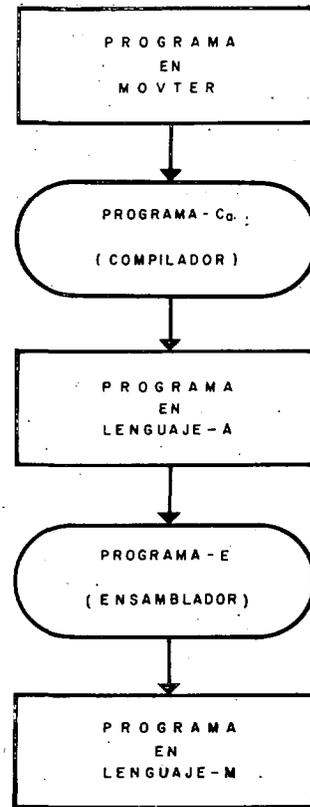


Figura 2.

De la misma forma se procesan los programas escritos en FORTRAN, ALGOL, COBOL, PL-I, LIST, etc. Cada uno de ellos precisa un programa C_{a_1} o C_{m_1} para poderse ejecutar en un determinado ordenador.

Limitaciones.

Supongamos que tenemos un programa X escrito en MOVTER y disponemos de un programa C_{a_1} que lo traduce al lenguaje A_1 del ordenador O_1 y adquirimos un segundo ordenador O_2 . Es evidente que para ejecutar el programa X en O_2 necesitaremos de un programa C_{a_2} del que no disponemos. Normalmente, la casa vendedora del O_2 nos habrá proporcionado compiladores (programas C) para algunos lenguajes de aplicación general como FORTRAN, COBOL, ALGOL, etc., pero no para uno tan especializado como el MOVTER, que además acaso incluso desconozca.

El escribir un programa C es costoso y difícil, necesita cientos, incluso miles de horas de

programador; por tanto, no es tarea que pueda emprenderse para un lenguaje de escasa aplicación. La decisión normal sería abandonar el MOVTER si el número de programas escritos en este lenguaje para O_1 y que tendremos que repetir en uno de los lenguajes de O_2 no nos compensa el trabajo de escribir este programa Ca_2 .

El problema no es normalmente tan simple; muy frecuentemente, sobre todo cuando O_1 y O_2 pertenecen a casas distintas, los lenguajes FORTRAN, ALGOL, etc., de cada sistema no son compatibles, y es muy general que ni tengan los mismos lenguajes.

Las casas constructoras de ordenadores se esfuerzan para que la transformación sea lo más fácil posible para sus clientes, manteniendo un cierto grado de compatibilidad entre los lenguajes simbólicos de sus ordenadores, con excepción de los lenguajes A. A veces también se proporciona esta facilidad por medio de un programa I (intérprete) que traduce un programa escrito en lenguaje A_1 a otro escrito en lenguaje M_2 . De esta forma, todos los programas escritos en *assembler* para O_1 pueden ejecutarse en O_2 .

Pero estas facilidades sólo remedian en muy pequeña proporción los problemas que presenta el mantenimiento del *software*. Hay bibliotecas inmensas de programas escritos en ALGOL que no pueden utilizar directamente los que utilizan el FORTRAN, y lo mismo ocurre al contrario.

Hemos puntualizado la potencia de los lenguajes especializados como el MOVTER, pero su eficacia queda restada por la dificultad de escribir su compilador.

Lenguajes C-C.

Veamos cómo se resuelven eficazmente estos problemas. Introduciremos para esto el lenguaje C-C. Este es un lenguaje O típico, en el que su tema especializado son los programas C; es, por tanto, en metalenguaje. No es preciso un lenguaje C-C único para todos los programas C; es más eficiente dividir los programas C por clases y tener un lenguaje C-C para cada clase. Un lenguaje C-C permite escribir programas C con la misma economía con que el FORTRAN permite escribir programas de tipo matemático. Examinemos el proceso de la figura 3 y el ejemplo de la figura 4. Obsérvese que

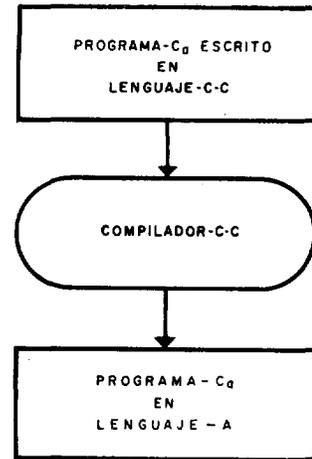


Figura 3.

el compilador C-C traduce a lenguaje A, y su utilidad se comprenderá si analizamos el caso anterior: paso del equipo O_1 al O_2 con un programa I, suministrado por la casa vendedora. Entonces nos bastará el proceso de la figura 5 para obtener el compilador MOVTER en lenguaje A_2 .

Si la casa vendedora no nos hubiese proporcionado el programa I, gracias a nuestro lenguaje C-C y su compilador, hubiésemos escrito con relativa facilidad el programa I en lenguaje C-C, y con el proceso descrito en las figuras 6 y 7 hubiésemos obtenido el programa I en la forma que lo necesitamos para el proceso de la figura 5. Con esto debe quedar claro que la principal dificultad que limita la eficacia de los lenguajes especializados queda

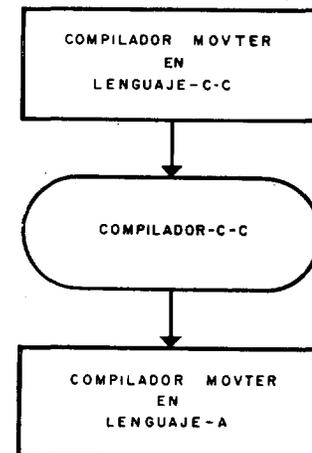


Figura 4.

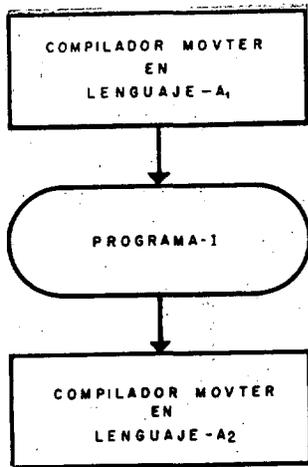


Figura 5.

vencida en gran medida y que, por tanto, se puede predecir un gran porvenir a los lenguajes O.

Todavía nos resta por describir otra gran aplicación del lenguaje C-C. Supongamos una empresa E productora de *software*. Esta empre-

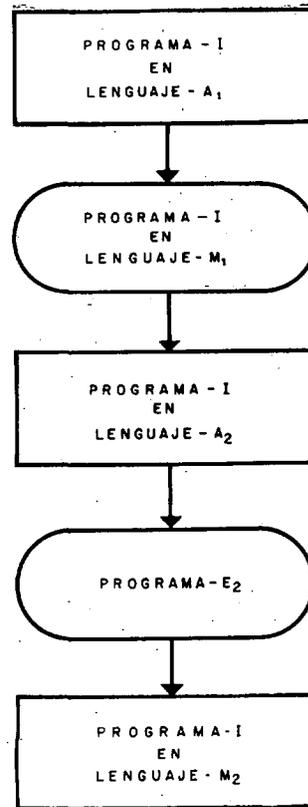


Figura 7.

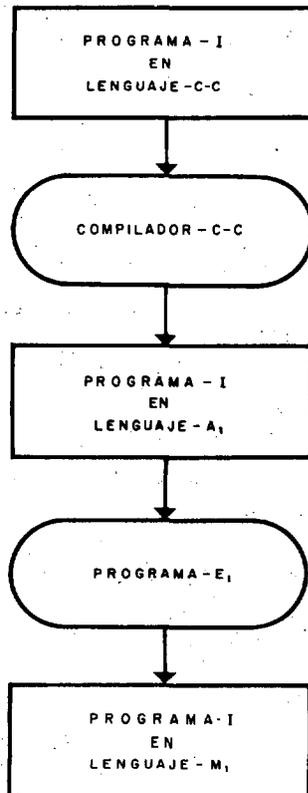


Figura 6.

sa, sirviéndose de su lenguaje o lenguajes C-C crea una serie de compiladores de la clase programas Ca, por medio de procesos del tipo descrito en la figura 3 para los lenguajes más usuales: FORTRAN, ALGOL, PL-I y BASIC.

Utilizando las librerías existentes escribe

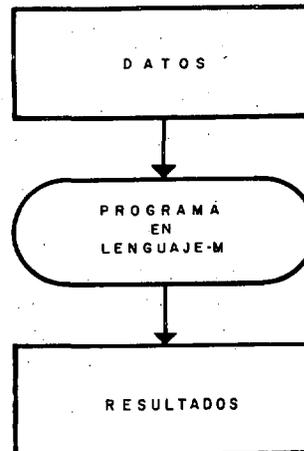


Figura 8.

programas de uso general a cinco niveles de sofisticación. Esto quiere resolver la imposibilidad real que existe de que los ordenadores pequeños puedan procesar programas que necesitan gran capacidad de memoria, velocidad de proceso y configuraciones muy completas. El sistema que proponemos es análogo a los ca-

tálogos de construcciones MECCANO, que pueden efectuarse con cada número. Esta empresa E puede ofertar a cualquier instalación *software* a niveles 1, 2, 3, 4 ó 5 casi inmediatamente, pues le basta obtener un programa I (si no lo tiene) para el equipo del cliente por medio del proceso ya explicado.

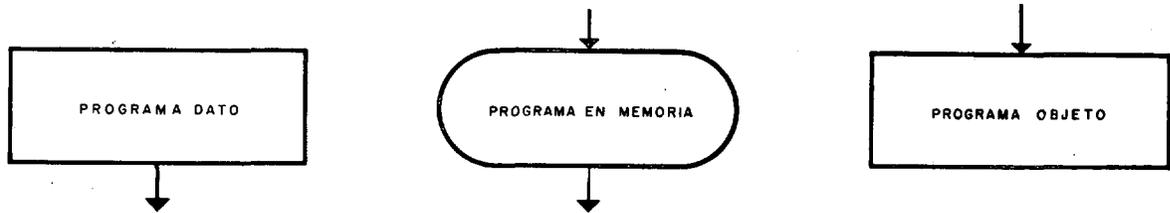


Figura 9.

Sumario.

Los lenguajes O constituyen una clase de lenguajes artificiales de importancia especial. En ellos se puede enunciar la resolución de problemas que se ejecutarán en ordenadores, lo que supone satisfacer el ideal de automatismo de la forma más perfecta conocida.

Los lenguajes O más utilizados son lenguajes, cuyo tema, aunque especializado, sea lo suficientemente general como para que el número de programas que se puedan escribir compense el trabajo que supone la construcción de un compilador.

Los lenguajes C-C vienen destinados a resolver, como su fin primero, esta dificultad, con lo que la aplicación de los lenguajes O adquirirá un enorme desarrollo. Pero no sólo tienen esta utilidad, sino que además permitirá construir bibliotecas realmente universales.

En breve exposición que hemos hecho eludimos de intento dificultades que se presentarían detallando los problemas de simular en un ordenador otro de configuración muy distinta. Como ésta hay otras dificultades de detalle, difíciles de vencer, que no hemos insinuado siquiera.

Sin embargo, esperamos que lo presentado despierte inquietudes en el lector imaginativo y le disponga a disfrutar de un mundo próximo en el que el hombre, creando el *software*, haga

un uso más intenso del instrumento más útil y fiel: el ordenador electrónico.

Bibliografía.

- Chapin: *Programming computers for business applications*.
- Leeds and Weinberg: *Computer programming fundamentals*.
- Rosen: *Programming systems and languages*.

GLOSARIO:

- Compilador C-C: Programa Ca que traduce el lenguaje C-C.
- Lenguaje A: Lenguaje O que se ensambla en un lenguaje M.
- Lenguaje C: Lenguaje de tema especializado sistemáticamente construido.
- Lenguaje C-C: Lenguaje O para escribir programas C.
- Lenguaje M: Lenguaje máquina.
- Lenguaje O: Lenguaje C ejecutable en un ordenador.
- Programa C: Programa compilador.
- Programa Ca: Programa C que traduce un lenguaje O a un lenguaje A.
- Programa C_m: Programa C que traduce un lenguaje O a un lenguaje M.
- Programa E: Programa traductor del lenguaje A al lenguaje M.
- Programa I: Programa traductor del lenguaje A₁ al lenguaje M₂.