



## EDITORIAL

**H**A sido nuestra constante preocupación el tema de la micro-computación en la educación porque hemos tenido la oportunidad, como también podríamos decir la suerte o el privilegio, de realizar experiencias señeras que siendo solamente validaciones relativamente modestas de lo que se ha hecho en los países líderes, son indicadores y argumentos muy contundentes que nos muestran el mundo maravilloso al que podrían tener acceso los educadores y educandos con la ayuda de los computadores ATARI.

La falta de información, como las dificultades que estamos pasando, junto con la falta de recursos de los establecimientos educacionales, conspiran fuertemente en contra de darle el verdadero cauce que merecen estas inquietudes, las cuales pueden tener repercusiones sociales incalculables.

Sabemos que el esfuerzo requerido para incorporar nuevos modelos y mejorar técnicas en la enseñanza es gigantesco. Pero también somos testigos de cómo hay educadores con gran vocación que al aceptar este desafío no han tenido más que permanentes satisfacciones.

El lenguaje LOGO es parte importantísima de este proceso y hemos aplaudido sin reserva la feliz participación de TELEDUC en difundirlo. Paralelamente ATARI también cuenta ya con un LOGO en español, que mantiene toda la capacidad de motivación que lo ha hecho famoso.

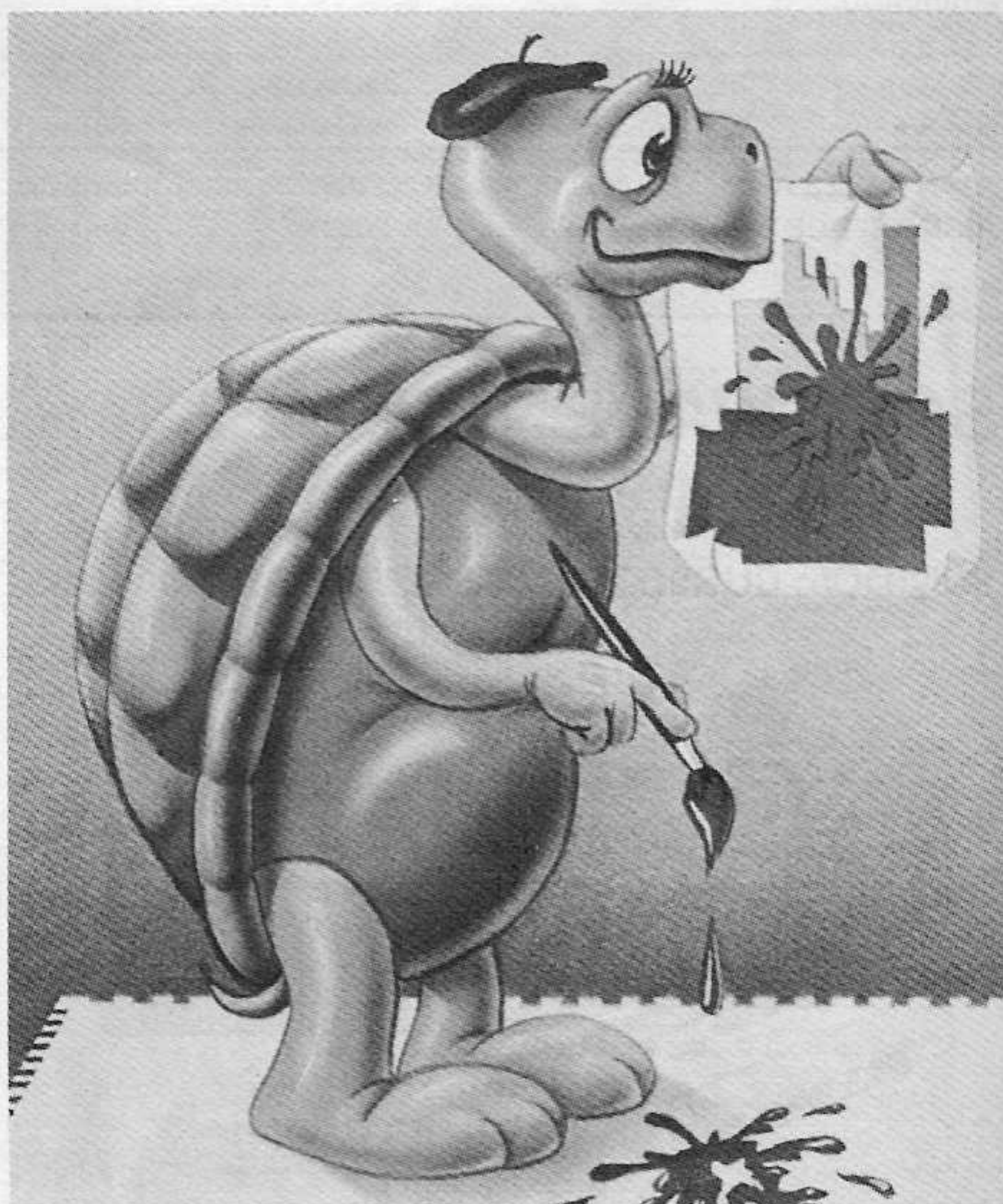
Nuestro BOLETIN comienza ahora un nuevo curso de LOGO, especialmente orientado a la versión en castellano. Comenzará con las instrucciones gráficas o de la geometría de la tortuga como se supone que es el modo más fácil y entretenido de ingresar al mundo de la computación. Por supuesto, veremos las TORTUGAS MULTIPLES con todas sus capacidades de cambiar de colores y formas. La tortuga dinámica con sus instrucciones permanentes de velocidad y también las instrucciones de bastones y perillas (Joystick y paddles) para que puedan hacer rápidamente un proyecto de juego.

Del manejo de memorias, condicionalidad, control de flujo, entregaremos los argumentos y ejercicios para que puedan entender fácilmente que significa un lenguaje estructurado de memoria dinámica de alto nivel y con esto proyectarse a cualquier lenguaje computacional.

El procesamiento de listas, también será un tema interesante que nos permitirá saber las ventajas del lenguaje LISP preferido de la inteligencia artificial.

Finalmente, nos ocuparemos de los informes por impresoras, como también de archivos de datos en cassettes y diskettes con lo cual podremos tener funcionando un sistema computacional completo bajo el lenguaje LOGO.

Como siempre, las cosas que se hacen por primera vez merecen muchos comentarios, porque son susceptibles de corregir y mejorar. Esperamos vuestra valiosa colaboración en el sentido de enviarnos comentarios cuando lo estimen necesario.





## HARDWARE DEL MES

### F - 80: IMPRESORA SMITH CORONA DE MATRIZ DE PUNTOS

Un interesante nuevo periférico compatible con su computador ATARI se ha incorporado al mercado, se trata de la impresora de matriz de puntos F-80 de la prestigiosa marca SMITH-CORONA.

Esta excelente impresora con una velocidad de impresión de 80 caracteres por segundo en forma bidireccional, imprime en formato normal de 80 columnas por línea o 132 caracteres condensados por línea. Dispone de capacidad gráfica de impresión con una resolución máxima de 576 puntos en 8 pulgadas, permitiendo gráficos de alta resolución en papel.



Por software se pueden controlar los diferentes tipos de impresión, como caracteres elongados, condensados, subrayados, escaciamiento entre líneas de 1/8, 1/6 ó n/72 de pulgadas por línea, escoger set de caracteres internacionales, saltos de página, etc.

La impresora dispone de una interfase de entrada paralela tipo centronics, que permite conectarse a un computador ATARI a través del módulo de interfase ATARI 850 y su correspondiente cable.

Y lo más importante, su reducido costo que la transforma en la solución más económica de impresión para su computador ATARI.

**\* programe  
su Atari  
con el profesor  
von Byte**



El manejo apropiado de string y el uso de archivos en cassettes en ATARI BASIC, comúnmente presenta algunas dificultades a los programadores que se inician. Por esta razón, y tomando como base la carta de consulta publicada este mes en la columna "BUZON ATARI", responderemos esta inquietud mediante un ejemplo que maneja un archivo de datos en un string y lo almacena en cassettes.

De acuerdo al ejemplo planteado, primero se dimensionarán adecuadamente las variables que contendrán los diferentes campos: nombre, edad, fecha de nacimiento y ciudad, como sigue:

```
NOMBRE$ (23)
EDAD$ (23)
FNAC$ (6)
CIUDAD$ (15)
```

Lo que arroja un largo total de registro de 46 bytes para cada persona, almacenado en la variable REGISTRO\$ (46). Asimismo, el contenido de todos los registros se almacenará en un gran string denominado ARCHIVO\$ (5000) que contendrá hasta un máximo de 108 registros diferentes. Todo el dimensionamiento de las variables se lleva a cabo en la línea 10 del programa ejemplo.

A continuación, en la línea 30 se rellena con espacios blancos toda la variable REGISTRO\$ y en la línea 40 se blanquea la variable ARCHIVO\$.

Luego se pasa a la subrutina 5000 para escoger la opción de leer un archivo previamente almacenado en cassette o crear uno nuevo.

La opción de lectura entrega los pasos a seguir para leer los datos desde el cassette, entre las líneas 5100 hasta 5240.

Al crear un archivo, se va a la rutina de ingreso de personas a partir de la línea 1000, almacenando cada uno de los campos en la variable REGISTRO\$ en las ubicaciones apropiadas, en las líneas 1080 a 1110, luego el registro completo se agrega a la cola de la variable ARCHIVO\$ en la línea 1112 y nuevamente se rellena de blancos el registro en la línea 1130, continuando este proceso para cada nuevo registro que se incorpore al archivo o terminando el ingreso según corresponda, en líneas 1170 a 1190.

Las opciones de modificar datos de persona o eliminar registro, consultan por el número de registro a actualizar, determinando su ubicación en el archivo con los punteros PI y PF.

(continuación...)

Al escoger la opción de término de trabajo, el programa va a la rutina 4000, donde se grabará el archivo completo al cassette, con las modificaciones efectuadas en las otras opciones.

Este ejemplo suministra un manejo apropiado de string en ATARI BASIC, como también la forma de utilizar archivos en cassette. Estudie detalladamente cada una de las rutinas de ingreso, actualización y eliminación que entrega el programa.

```

2 REM * Manejo ARCHIVO en CASSETTE *
4 REM * BOLETIN CENTRO ATARI *
6 REM * ENERO 1985 *
8 REM *****
10 DIM RESPUESTA$(1), LETRA$(1), ARCHIVO
$(5000), REGISTRO$(46), NOMBRE$(23), EDAD
$(2), FNAC$(6), CIUDAD$(15)
30 REGISTRO$=" ":REGISTRO$(46)=" ":REG
ISTRO$(2)=REGISTRO$
40 ARCHIVO$=""
90 GOSUB 5000
100 REM Construye Menu Principal
110 GRAPHICS 0:POKE 53774,64:POKE 16,6
4
120 TRAP 110:POSITION 1,3:? "MENU PRIN
CIPAL"
130 POSITION 1,5:? "1- INGRESAR PERSO
NAS"
140 POSITION 1,7:? "2- MODIFICAR PERSO
NAS"
150 POSITION 1,9:? "3- ELIMINAR PERSO
NAS"
180 POSITION 1,11:? "4- TERMINAR"
190 POSITION 1,20:? "ELIJA UN NUMERO "
:INPUT NUMERO
200 IF NUMERO<1 OR NUMERO>4 THEN ? CHR
$(253):GOTO 110
210 TRAP 40000:ON NUMERO GOTO 1000,200
0,3000,4000
1000 REM Ingresa personas
1004 NP=NP+1:IF NP>108 THEN GOTO 1500
1010 GRAPHICS 0:POKE 53774,64:POKE 16,
64
1011 GOSUB 7000
1012 POSITION 10,2:? "INGRESAR PERSONA
S"
1020 POSITION 1,4:? "DEME NOMBRE " :I
NPUT NOMBRE$
1030 POSITION 1,6:? "DEME E D A D " :I
NPUT EDAD$
1040 POSITION 1,8:? "DEME F. NAC. " :I
NPUT FNAC$
1050 POSITION 1,10:? "DEME CIUDAD " :I
NPUT CIUDAD$
1060 POSITION 1,20:? "DESEA CORREGIR D
ATOS S/N " :INPUT RESPUESTA$
1070 IF RESPUESTA$="S" THEN GOTO 1010
1072 IF RESPUESTA$(">")"N" THEN GOTO 1060

1080 REGISTRO$(1,23)=NOMBRE$
1090 REGISTRO$(24,25)=EDAD$

```

```

1100 REGISTRO$(26,31)=FNAC$
1110 REGISTRO$(32,46)=CIUDAD$
1112 ARCHIVO$(LEN(ARCHIVO$)+1)=REGISTR
O$
1130 REGISTRO$=" ":REGISTRO$(46)=" ":R
EGISTRO$(2)=REGISTRO$
1170 POSITION 1,20:? "DESEA INGRESAR O
TRA PERSONA S/N " :INPUT RESPUESTA$
1180 IF RESPUESTA$="N" THEN GOTO 110
1190 GOTO 1000
1500 REM ARCHIVO lleno
1510 GRAPHICS 0:POKE 53774,64:POKE 16,
64
1520 POSITION 10,10:? CHR$(253);"ARCHI
VO LLENO"
1530 FOR PAUSA=1 TO 400:NEXT PAUSA:GOT
O 110
2000 REM Modifica personas
2020 GRAPHICS 0:POKE 53774,64:POKE 16,
64
2030 TRAP 2020:? "NUMERO PERSONA A MOD
IFICAR " :INPUT NMOD
2040 TRAP 40000:FOR X=1 TO NP
2042 IF NMOD=X THEN GOTO 2050
2044 NEXT X
2046 POSITION 1,20:? CHR$(253);" PE
RSONA NO EXISTE"
2048 FOR PAUSA=1 TO 400:NEXT PAUSA:GOT
O 2170
2050 PI=1+46*(X-1):PF=PI+7
2060 REGISTRO$=" ":REGISTR$=ARCHIVO$(P
I,PF)
2070 GRAPHICS 0:POKE 53774,64:POKE 16,
64
2080 POSITION 10,2:? "MODIFICA PERSONA
S"
2082 POSITION 1,4:? "PERSONA # " :X
2090 POSITION 1,6:? "NOMBRE " :REGISTRO
$(1,23)
2100 POSITION 1,8:? "EDAD " :REGISTRO
$(24,25)
2110 POSITION 1,10:? "F.NAC. " :REGISTR
O$(26,31)
2120 POSITION 1,12:? "CIUDAD " :REGISTR
O$(32,46)
2130 POSITION 1,20:? "MODIFICA S/N " :I
NPUT RESPUESTA$
2140 IF RESPUESTA$="N" THEN GOTO 2170
2150 IF RESPUESTA$="S" THEN GOTO 2202
2160 GOTO 2130
2170 POSITION 1,20:? "MODIFICA OTRA PE
RSONA S/N " :INPUT RESPUESTA$
2180 IF RESPUESTA$="N" THEN GOTO 110
2190 IF RESPUESTA$="S" THEN GOTO 2020
2200 GOTO 2170
2202 GRAPHICS 0:POKE 53774,64:POKE 16,
64
2203 GOSUB 7000:REGISTRO$=" ":REGISTRO
$(46)=" ":REGISTRO$(2)=REGISTRO$
2204 POSITION 10,2:? "MODIFICA PERSONA
S"
2220 POSITION 1,4:? "DEME NOMBRE " :IN
PUT NOMBRE$
2230 POSITION 1,6:? "DEME EDAD " :IN
PUT EDAD$
2240 POSITION 1,8:? "DEME F.NAC. " :IN
PUT FNAC$

```

```

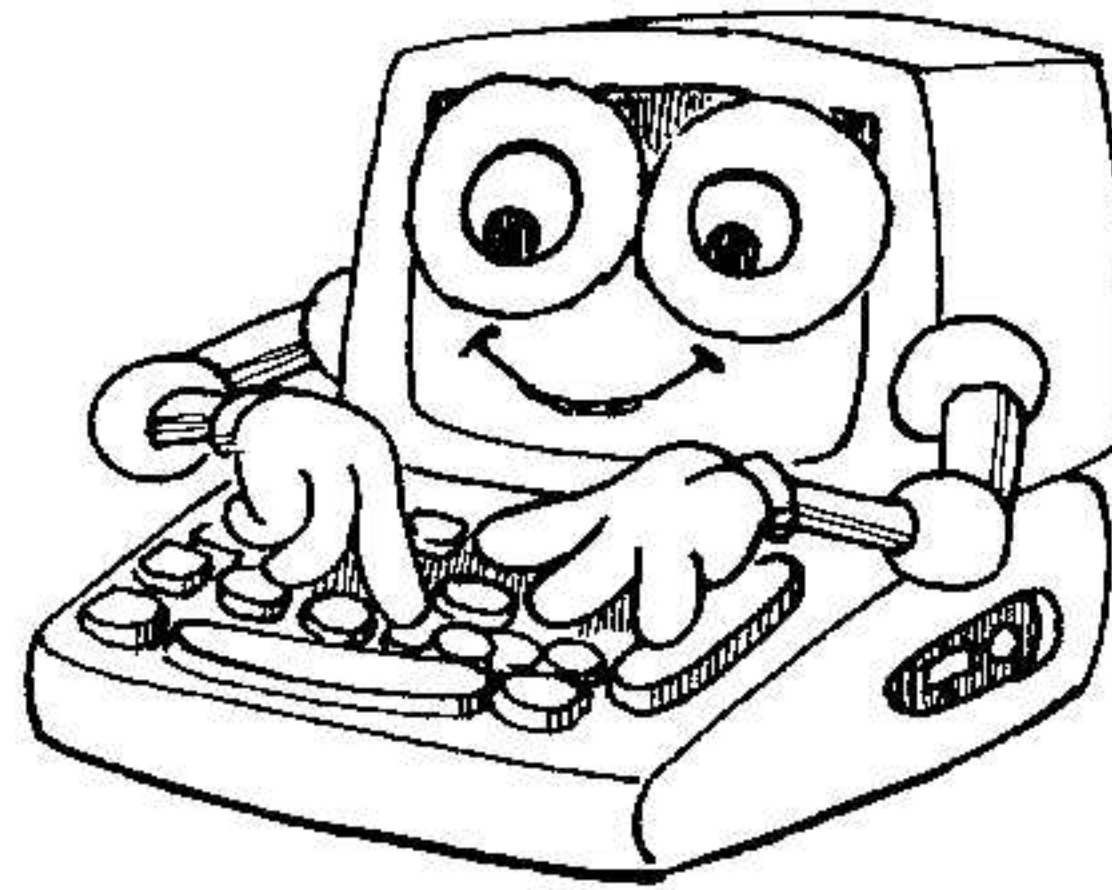
2250 POSITION 1,10:? "DEME CIUDAD ";:I
NPUT CIUDAD$
2260 POSITION 1,20:? "DESEA CORREGIR D
ATOS S/N ";:INPUT RESPUESTA$
2270 IF RESPUESTA$="S" THEN GOTO 2202
2280 IF RESPUESTA$="N" THEN GOTO 2300
2290 GOTO 2260
2300 REGISTRO$(1,23)=NOMBRE$
2310 REGISTRO$(24,25)=EDAD$
2320 REGISTRO$(26,31)=FNAC$
2330 REGISTRO$(32,46)=CIUDAD$
2340 ARCHIVO$(PI,PF)=REGISTRO$
2350 REGISTRO$=" ":REGISTRO$(46)=" ":R
EGISTRO$(2)=REGISTRO$
2360 GOTO 2170
2400 GRAPHICS 0:POKE 53774,64:POKE 16,
64
2410 POSITION 6,12:? "NO HAY PERSONAS
EN ARCHIVO"
2420 FOR PAUSA=1 TO 400:NEXT PAUSA
2430 GOTO 110
3000 REM Elimina persona
3020 GRAPHICS 0:POKE 53774,64:POKE 16,
64
3030 ? "NUMERO PERSONA A ELIMINAR ";:I
NPUT NMOD
3040 FOR X=1 TO NP
3050 IF NMOD=X THEN GOTO 3090
3060 NEXT X
3070 POSITION 1,20:? CHR$(253);" PERSO
NA NO EXISTE "
3080 FOR PAUSA=1 TO 400:NEXT PAUSA:GOT
O 3150
3090 REM
3100 PI=1+46*(X-1):PF=PI+45
3102 IF LEN(ARCHIVO$)=46 THEN ARCHIVO$
="":GOTO 3130
3104 IF PI=1 THEN ARCHIVO$=ARCHIVO$(PF
+1):GOTO 3130
3106 IF NMOD=NP THEN ARCHIVO$=ARCHIVO$
(1,PF-46):GOTO 3130
3120 ARCHIVO$(LEN(ARCHIVO$(1,PI-1))+1)
=ARCHIVO$(PF+1)
3130 NP=NP-1:POSITION 1,20:? CHR$(253)
;"PERSONA ELIMINADA"
3140 FOR PAUSA=1 TO 400:NEXT PAUSA
3150 POSITION 1,20:? "DESEA ELIMINAR O
TRA PERSONA S/N ";:INPUT RESPUESTA$
3160 IF RESPUESTA$="S" THEN GOTO 3020
3170 IF RESPUESTA$="N" THEN GOTO 110
3180 GOTO 3150
4000 REM Termino del trabajo
4010 GRAPHICS 0:POKE 53774,64:POKE 16,
64
4020 POSITION 10,1:? "GRABACION DE ARC
HIVO"
4030 POSITION 6,4:? "PASOS A SEGUIR:"
4040 POSITION 3,6:? "1- Rebobine CINTA
"
4050 POSITION 3,7:? "2- Presione Tecla
s RECORD Y PLAY de la GRABADOR
A"
4060 POSITION 3,9:? "3- Presione 2 vec
es tecla RETURN"
4070 POSITION 3,13:? "Presione RETURN
cuando este listo";:INPUT RESPUESTA$

```

```

4080 POSITION 10,20:? "GRABANDO ARCHIV
O"
4090 OPEN #1,8,0,"C:"
4100 FOR X=1 TO NP
4110 PI=1+46*(X-1):PF=PI+45
4120 REGISTRO$="":REGISTRO$=ARCHIVO$(P
I,PF)
4130 ? #1;REGISTRO$
4140 NEXT X:GRAPHICS 0:END
5000 REM Genera o Lee Archivo en CASSE
TTE
5010 GRAPHICS 0:POKE 53774,64:POKE 16,
64
5012 POSITION 13,2:? "OPCIONES"
5020 POSITION 1,4:? "L- LEER ARCHIVO D
ESDE CASSETTE"
5030 POSITION 1,6:? "C- CREAR ARCHIVO
EN CASSETTE"
5040 POSITION 1,8:? "M- MENU PRINCIPAL
"
5050 POSITION 4,20:? "ESCOJA UNA LETRA
";:INPUT LETRA$
5060 IF LETRA$="L" THEN GOTO 5100
5070 IF LETRA$="C" THEN POP :GOTO 1000
5080 IF LETRA$="M" THEN RETURN
5090 GOTO 5010
5100 REM Lee ARCHIVO desde CASSETTE
5110 GRAPHICS 0:POKE 53774,64:POKE 16,
64
5120 POSITION 10,1:? "LECTURA DE ARCHI
VO"
5130 POSITION 6,4:? "PASOS A SEGUIR:"
5140 POSITION 3,6:? "1- Rebobine CINTA
"
5150 POSITION 3,7:? "2- Presione Tecla
PLAY de la GRABADOR
A"
5160 POSITION 3,9:? "3- Presione 2 vec
es tecla RETURN"
5170 POSITION 3,13:? "Presione RETURN
cuando este listo";:INPUT RESPUESTA$
5180 POSITION 10,20:? "LEYENDO ARCHIVO
"
5190 OPEN #1,4,0,"C":ARCHIVO$="":NP=0
5200 TRAP 5220
5210 REGISTRO$="":INPUT #1;REGISTRO$:N
P=NP+1
5212 ARCHIVO$(LEN(ARCHIVO$)+1)=REGISTR
O$:GOTO 5210
5220 TRAP 40000:IF PEEK(195)=136 THEN
CLOSE #1:RETURN
5240 POP
6000 REM Rutina de ERROR
6010 GRAPHICS 0:? CHR$(253)
6012 POSITION 14,2:? "ADVERTENCIA"
6020 POSITION 4,7:? "Lo lamento pero U
d. debe crear el ARCHIVO,"
6030 POSITION 4,9:? "de lo contrario n
o podra seguir adelante."
6040 POSITION 4,20:? "Presione RETURN
para continuar";:INPUT RESPUESTA$
6050 GOTO 90
7000 REM Limpia Variables
7010 NOMBRE$="":EDAD$="":FNAC$="":CIUD
AD$="":RETURN

```



En la columna de colaboración de este mes, hemos escogido un interesante aporte enviado por el señor Patricio Ruiz, casilla 26, correo 34 de Santiago, quien demuestra una interesante investigación y conocimiento del computador ATARI 600 XL, entregando sus comentarios y una interesante subrutina en lenguaje de máquina a todos nuestros lectores.

## \* aquí opinan los lectores de Atari

El primer comentario se refiere a la escritura en pantalla de los programas en lenguaje de máquina. En efecto, más de algún usuario habrá sorprendido que al hacer funcionar algunos de los juegos programados en lenguaje de máquina en la revista COMPUTE!, y que se cargan con el programa especialmente diseñado para este objeto, MLX, no escriben correctamente en pantalla los títulos y escrituras. Un caso típico de este problema ocurre en los programas titulados SHOOT y CHOPPEROID. La causa del problema está en que estos programas fueron diseñados para ser operados en el modelo antiguo, el ATARI 400/800, que tiene el puntero del hardware para transformación de código de caracteres de ATASCII a Interno, ubicado en una posición diferente que en el nuevo modelo 600 XL. En efecto, este puntero está ubicado en la posición \$FB49 (64329 dec.) en el modelo 600 XL. En consecuencia, para corregir el problema de escritura bastará con cambiar en la subrutina correspondiente, donde dice 246,254 se debe poner 073,251.

La manera de efectuar este cambio es bastante simple, bastará con cargar el programa en cuestión usando el programa de carga MXL, escribir nuevamente las líneas corregidas y volver a grabar en cassette o diskette el programa corregido. En los programas antes mencionados, las líneas a corregir son las siguientes:

SHOOT 5152: 073, 251, 168, 104, 170, 152, 182  
CHOPPEROID 4790: 159, 029, 073, 251, 168, 104, 198

El segundo comentario se refiere al uso de subrutinas de hardware para mover bloques de memoria de un lugar a otro. En efecto, para la compilación de programas en BASIC, el hardware dispone de subrutinas muy eficientes que permiten correr bloques de memoria de un lugar a otro ya sea para abrir espacios necesarios para nuevas líneas de programa que se intercalan entre las existentes, o para suprimir espacios correspondientes a líneas de programa que se eliminan. En el modelo ATARI 600 XL, estas subrutinas están ubicadas en las siguientes direcciones del hardware: en \$A87A (43130 dec.) la que abre espacios y en \$A8F7 (43255 dec.) la que suprime espacios.

Estas subrutinas pueden ser usadas también por el programador que desee mover bloques de datos en memoria de un lugar a otro. Quizás la aplicación más corriente es la de traer la lista de caracteres desde el ROM hacia el RAM para ser usados, con algunas modificaciones, como elementos de animación. En

este caso, el problema consiste en mover un bloque de datos de 4 páginas, ó 1024 bytes, desde la página 224 en ROM hacia una ubicación predefinida en RAM. Si bien es cierto que esto se puede hacer usando lenguaje BASIC, resulta muchísimo más rápido hacerlo usando la infraestructura de la subrutina de hardware recién mencionadas. En esta aplicación se utiliza

la subrutina que corre segmentos de memoria hacia abajo; es decir, la que BASIC emplea para suprimir espacios desusados.

Para la aplicación práctica, bastará con definir la dirección en la memoria RAM hacia donde se traerá la lista de caracteres y denominarla START y luego hacer un llamado a la subrutina del hardware por medio del comando USR.

Algunos casos típicos para la ubicación de START pueden ser los siguientes:

- a)  $START = (PEEK(106) - 12) * 256$
- b)  $START = (PEEK(106) - 16) * 256$
- c)  $POKE 106, PEEK(106) - 5 : START = (PEEK(106) + 1) * 256$

En los casos a) y b) se elige una ubicación desplazada en 12 ó 16 páginas desde el tope del RAM, dependiendo del modo gráfico en uso, para copiar la lista de caracteres fuera del área de la memoria en pantalla. En el caso c) se corre el tope del RAM en 5 páginas hacia abajo para dejar un espacio donde ubicar la lista de caracteres protegido de interferencias con el programa.

La llamada a la subrutina de traslado se hace con la siguiente instrucción:

```
Z=USR(ADR("h") [ ] [E] [Z] h" [D] [ ] [D] [Y] [T] [W] [U] [X] hLS) ",
START)
```

debe tenerse especial cuidado al copiar esta instrucción por cuanto cualquier error puede causar el bloqueo del computador. Los símbolos entre paréntesis [ ] deben escribirse con la tecla CONTROL oprimida y los símbolos subrayados deben escribirse en modo inverso, es decir, con la tecla ATARI.

Finalmente, y sólo a título informativo, debe destacarse que existe un error en una de las subrutinas de hardware recién descritas.

En efecto, la subrutina que mueve bloques de datos hacia arriba en la memoria contiene un error que se hace presente cuando un bloque de datos a mover tiene una longitud igual a un número entero de páginas. Afortunadamente esto es poco usual pero debe tenerse precaución al usar esta subrutina en forma directa. La manera de corregir este error sería de reemplazar el número 10 que existe en la posición \$A8E1 (43233 dec.) por un número 1.

Lamentablemente esto no lo puede hacer el usuario, sino que debería ser corregido por el fabricante.

# ATARI®

## EN LA EDUCACION



A fin de conocer las diversas actividades desarrolladas en la Quinta Región por el Centro de Enseñanza Asistida por Computador CEAC, solicitamos al Ingeniero en Sistemas, Sr. Marcos Oyanedel Núñez nos hablara sobre el centro que dirige.

*Nuestro Centro fue creado en marzo de 1984, como resultado de la inquietud compartida por un grupo de Ingenieros en Sistemas de Información de la U.T.F.S.M., quienes junto a un grupo de pedagogos de diferentes especialidades visualizaron el amplio campo que están abriendo los computadores en nuestro Sistema Educacional, ya sea como máquinas de calcular, máquinas de enseñar o como medios intelectuales creativos.*

*Es así como durante este breve período hemos podido consolidar un equipo multidisciplinario, cuyos principales objetivos han estado orientados hacia la difusión a nivel regional de la computación en el campo educativo, enseñanza de aplicaciones, enseñanza de lenguajes, preparación con ayuda del computador de alumnos para rendir P.A.A., e investigación y desarrollo de aplicaciones.*

*En lo relativo a la difusión, nuestro propósito ha estado orientado hacia que la comunidad escolar (Directores de colegios, Profesores, Orientadores, Alumnos) puedan visualizar las enormes proyecciones que tiene el computador como una herramienta de apoyo en el área educativa. Así es como durante el año recién pasado, dimos charlas en radioemisoras locales, se han realizado visitas a diversos establecimientos educacionales donde alumnos y profesores han tenido la oportunidad de interactuar durante algunas horas con las diversas modalidades que ofrece un microcomputador.*

*Además, profesores y alumnos han respondido gentilmente a nuestra invitación, visitando el centro y han podido apreciar en el terreno mismo las ventajas y facilidades que otorgamos para incorporar a niños, jóvenes y adultos al campo de la computación.*

*El área de la enseñanza está dividida en dos niveles: el de la enseñanza de aplicaciones, que está dirigida a todas aquellas personas (ejecutivos, profesionales, docentes, etc.) que dentro de su quehacer requieran comprender y poder aplicar alguna de las modalidades específicas de trabajo con un computador.*

*Así es como, por ejemplo, se dictan cursos de Procesamiento de Textos, Visicalc, Bases de Datos y otros. El otro nivel está dirigido para que diferentes personas puedan aprender y aplicar algún lenguaje computacional como: LOGO, BASIC, PILOT, COBOL, PASCAL, etc. Dependiendo de la edad cronológica (se parte con niños de seis años de edad) y del grado de conocimiento de la materia. Esta actividad se desarrolla mediante la dictación de cursos personalizados que respondan a los intereses particulares de cada alumno.*

*En el área de investigación se puso en marcha un proyecto en conjunto con el Departamento de Informática de la U.T.F.S.M., donde tienen la oportunidad de trabajar profesores y alumnos de diferentes cursos de las carreras de Ingeniería Civil y Ejecución en Informática junto a los profesionales del centro, en el desarrollo y evaluación de software educativo. Esperamos que durante el presente año podamos empezar a dar a conocer lo desarrollado en esta área, la cual consideramos como muy importante dado el impacto que están teniendo y llegarán a tener los microcomputadores en la sociedad.*

CEAC, se encuentra ubicado en Bellavista 280, Agua Santa, de la ciudad de Viña del Mar.

## CENTRO ATARI®

Adquiera en los **CENTROS ATARI** a lo largo del país todo el material necesario para su Computador ATARI.

Todo el hardware, software, accesorios y literatura relacionada, podrá encontrarlos en su **CENTRO ATARI** más cercano.

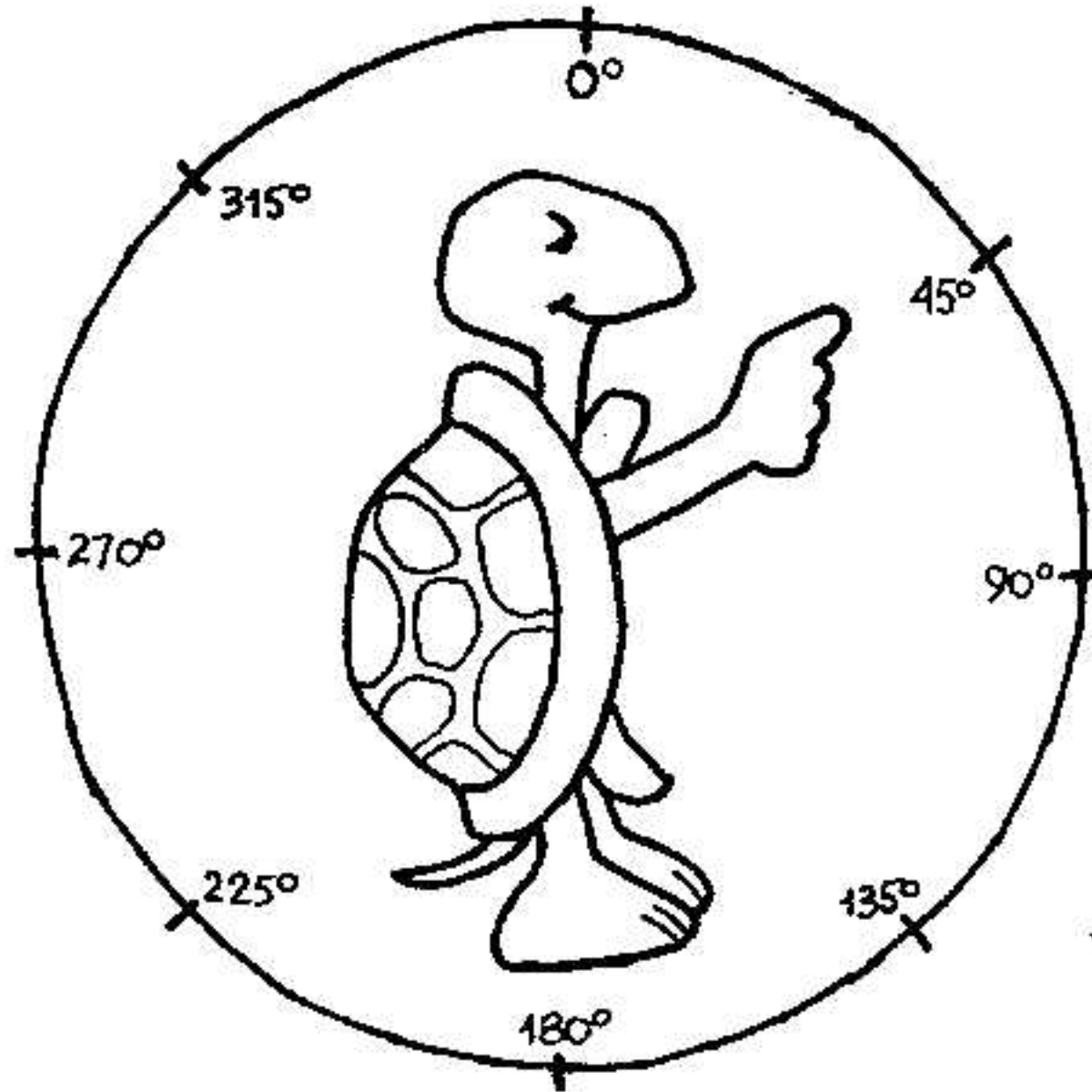
**CENTRO ATARI**  
un mundo para su Computador ATARI

- HARDWARE
- SOFTWARE
- ACCESORIOS
- LITERATURA (libros y revistas)
- CURSOS
- ASISTENCIA

**CENTROS ATARI en:**

ARICA — IQUIQUE — ANTOFAGASTA —  
SANTIAGO — SAN ANTONIO — VIÑA  
DEL MAR — CURICO — CONCEPCION —  
TEMUCO — OSORNO — PUNTA ARENAS

## CURSO LOGO EN ESPAÑOL



Durante el desarrollo del curso revisaremos las siguientes materias:

**GEOMETRIA DE LA TORTUGA**  
**EDICION DE TAREAS**  
**VARIABLES LOCALES**  
**MANEJO DE MEMORIA**  
**EDITOR DE FORMAS**  
**CONDICIONALES**  
**OPERACIONES ARITMETICAS**  
**OPERACIONES LOGICAS**  
**BASTONES Y MANDOS**  
**MANEJO DE LISTAS**  
**ARCHIVOS**  
**INFORMES POR IMPRESORA**

## GEOMETRIA DE LA TORTUGA

La geometría de la tortuga pretende un encuentro natural del niño con los números a muy temprana edad. Este encuentro natural tiene dos significados:

- Que los números sean una necesidad real del niño.
- Que la geometría inherente sea una geometría incremental relativa a la posición del sujeto para que tenga validez universal, más que la geometría de Euclides o de Descartes, y aplique sin aparente contrasentido en el cálculo diferencial e integral.

Transformar los números en necesidad es un problema del ambiente sugestivo del entorno como también de los elementos de motivación implícitos, en el lenguaje LOGO. Los elementos de motivación que incluye el ATARI LOGO son:

- Cuatro tortugas programables que pueden cambiar su color entre 126 colores y su forma entre 16 formas distintas.
- Instrucciones permanentes de velocidad y condicionalidad.
- Todo tipo de detección de colisiones.
- Sonidos.

La geometría misma se basa en el desplazamiento de la tortuga sobre la pantalla del televisor o monitor, donde puede dejar un trazo de distintos colores, borrarlos y redibujarlos.

Las instrucciones básicas son de desplazamiento y giro que definen el estado de la tortuga en los atributos de posición, dirección y sentido.

**AV,** (avanza)  
**RE,** (retrocede)  
**DE,** (gira a la derecha)  
**IZ,** (gira a la izquierda)

Estas instrucciones deben ir acompañadas de un número que indique las unidades de desplazamiento o giro.

El desplazamiento se mide en pasitos de tortuga, que son muy cortitos. La pantalla del ATARI LOGO tiene 320 pasitos de ancho y 240 de alto.

El giro se mide en grados de tal forma que un giro completo por la derecha se consigue con la instrucción DE 360.

Para dibujar existen varias instrucciones que sirven para que la tortuga maneje sus lápices. En primer término veremos sólo tres.

**LA** Indica a la tortuga bajar el lápiz  
**SL** Indica a la tortuga subir el lápiz  
**LG** Indica a la tortuga usar el lápiz-goma

Nuestro primer encuentro con el LOGO lo hemos estimado mejor hacerlo con la tortuga, sin embargo el LOGO comienza siempre con una imagen en la pantalla dedicada al TEXTO. Para llegar hasta la tortuga y su campo de acción es necesario dar cualquier instrucción de manejo de gráfico. Si utilizamos la instrucción TG, que indica a la tortuga que aparezca, se cambia la pantalla de TEXTO por la pantalla gráfica y aparece la TORTUGA en el centro de la pantalla.

La tortuga llega al centro de la pantalla, cada vez que se la llama por primera vez, con su lápiz listo para dibujar, es decir, cualquier desplazamiento que se le ordene irá dejando un trazo por donde ha pasado.

Con estas ocho instrucciones debe comenzar la familiarización con los elementos que componen un sistema computacional.

Las instrucciones de estado de la tortuga (AV, RE, DE, IZ) deben ir acompañadas de un número que da la relación del nuevo estado.

Por ejemplo:

**AV 90**

El LOGO es muy celoso al interpretar una orden. El espacio que separa AV de 90 es clave que indica que AV es un elemento de la instrucción y 90 es otro elemento.

Con la instrucción AV 90 queremos que la tortuga avance 90 pasitos en la dirección o rumbo que actualmente apunta. Veamos un ejercicio real:

### EJERCICIO 1.

- Con el computador debidamente conectado a su monitor (ver manual del computador o de cartridges de LOGO ATARI en español) insertar el cartridge del LENGUAJE LOGO.
- Encienda el TV o monitor y el computador.
- Luego aparecerá en pantalla el mensaje:  
**BIENVENIDO A ATARI LOGO**
- Tipée TG y luego presione la tecla RETURN

**NOTA IMPORTANTE:** La tecla RETURN sirve para indicar a LOGO que nuestro mensaje está completo, y es el turno de él para interpretar el mensaje y realizar las instrucciones.

Es la tecla que más se utiliza y de aquí en adelante se supondrá que cada línea de mensaje termina con presionar dicha tecla. Si la línea del mensaje es más larga que la línea física, se puede seguir escribiendo aún habiendo llegado el cursor al lado derecho de la



pantalla, pues automáticamente salta al extremo de la línea siguiente. Las líneas que exceden el ancho de la pantalla disponible quedan marcadas con una flechita → que indica que continúa en la siguiente línea física.

5. En el centro de la pantalla debe aparecer la tortuga apuntando hacia arriba. Nuestro primer dibujo será un cuadrado. Digite:

AV 50.  
DE 90.  
AV 50.  
DE 90.  
AV 50.  
DE 90.  
AV 50.  
DE 90.

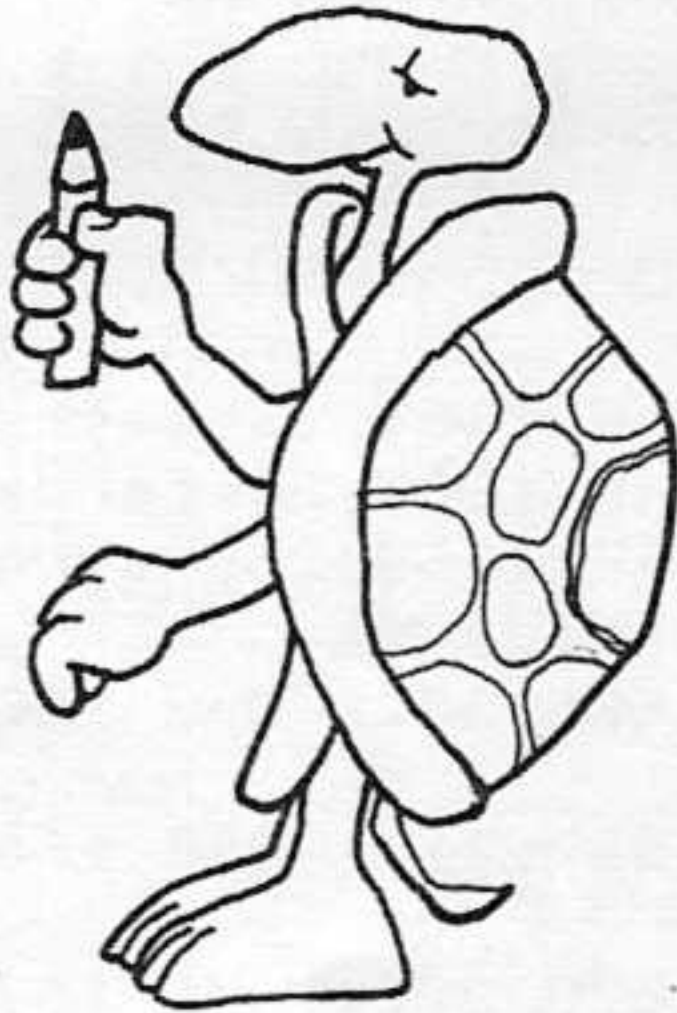
Cada una de estas instrucciones es ejecutada al presionar RETURN.

Finalmente se completó el cuadrado gracias a que la tortuguita apareció con su lápiz listo para dibujar.

Recordemos que la filosofía LOGO implica disciplinas y hábitos en orden a optimizar las tareas cognitivas y finalmente la creatividad.

El conocimiento de la realidad lo podemos aplicar durante todo el presente curso, y es el momento de empezar.

Con las instrucciones que hemos presentado le vamos a tomar el sentido a la geometría LOGO. Nuestro interés, imaginación y creatividad deben ser los que nos van a plantear el próximo ejercicio.

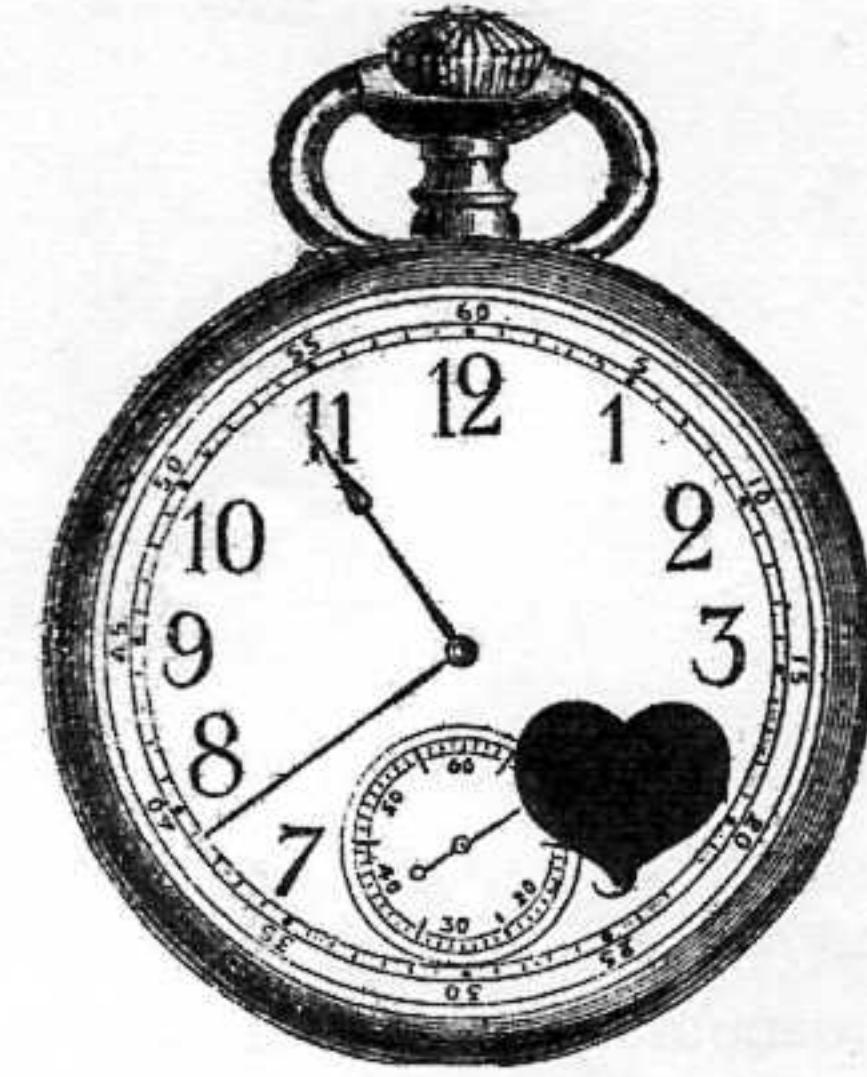


## EJERCICIO 2.

Desarrollar una o varias ideas sencillas de geometría y realizarlas. Recomendamos un paisaje, casas, árboles, montañas, nubes, sol, etc., para los casos en que no se tenga una inspiración propia. Se pretende generar una experiencia semejante a la que tendría un niño con sus instintos relativos al conocimiento, totalmente vitales, activos y sin distorsiones.

- Explorar el computador
- Descubrir relaciones causa-efecto
- Manejar estas relaciones causa-efecto para desarrollar una idea propia
- Sentir el placer de lograr manejar esta realidad
- Terminar el proyecto.

OJO: Puede borrar toda la imagen con la instrucción LM.



## CONSTRUYA UN RELOJ DE TIEMPO REAL

El Computador ATARI dispone de un reloj interno, que constantemente se está actualizando, ubicado en las posiciones de memoria 18, 19 y 20 decimales.

A continuación se entrega una subrutina que permite controlar las horas, minutos y segundos, utilizando estas posiciones de memoria. La rutina de cálculo de horas comienza en la línea 160 y usted puede incorporarla a sus propios programas.

```

10 REM * RUTINA DE RELOJ *
20 REM * BOLETIN CENTRO ATARI *
30 REM * ENERO - 1985 *
40 REM *****
100 GRAPHICS 2+16
110 POKE 20,0:POKE 19,0:POKE 18,0
120 GOSUB 170
130 POSITION 4,0:? #6;"RELOJ ATARI"
140 POSITION 6,5:? #6;HRS;" ":";MIN;" ":";
SEG;" "
154 GOTO 120
160 REM Rutina del reloj
170 TIEMPO=(PEEK(20)+PEEK(19)*256+PEEK
(18)*256*256)/60
180 REM Calcula segundos
190 SEG=INT((TIEMPO/60-INT(TIEMPO/60))
*60)
200 REM Calcula minutos
210 MIN=INT((TIEMPO/3600-INT(TIEMPO/36
00))*60)
220 REM Calcula horas
230 HRS=INT((TIEMPO/216000-INT(TIEMPO/
216000))*60)
240 RETURN

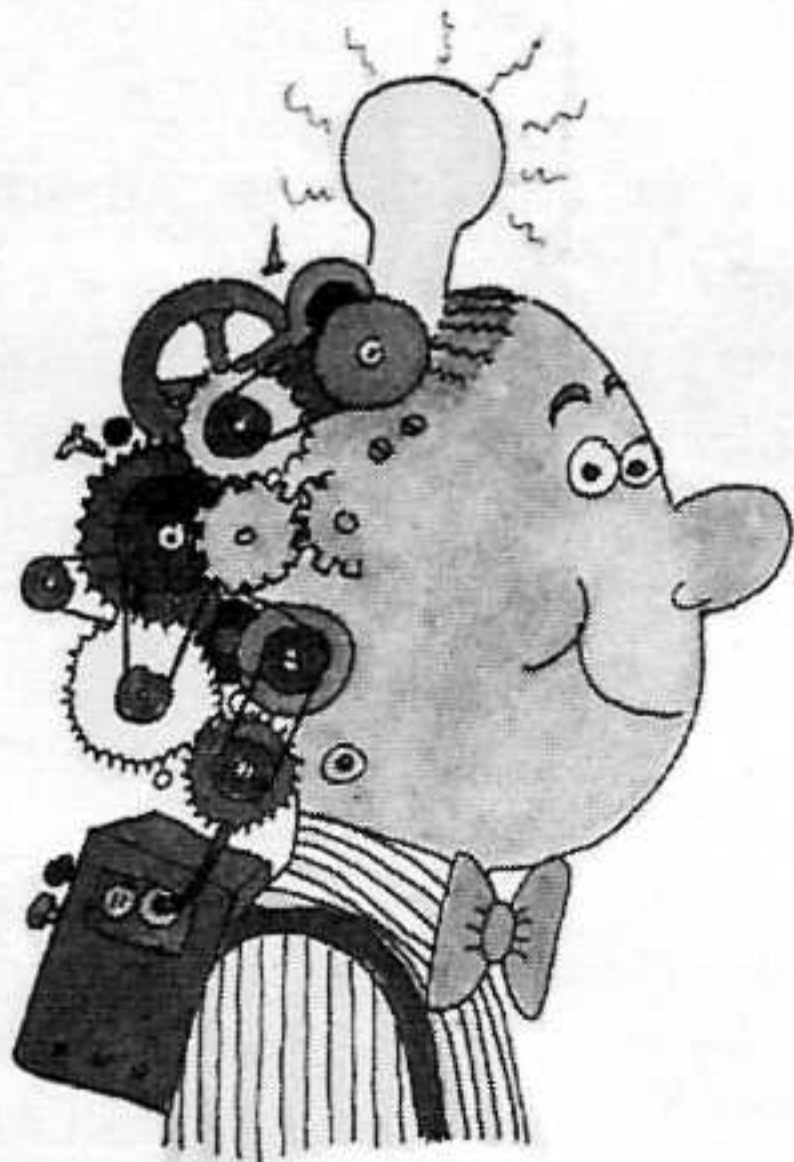
```

## IDEAS Y TRUCOS

Para los entusiastas de las capacidades gráficas y de colores de su computador ATARI, se entrega en esta sección una rutina en lenguaje de máquina que genera una cascada de colores sobre una figura dada. La rutina que se encuentra entre las líneas 30.000 a 30.100 puede ser incorporada a cualquier programa BASIC para producir el efecto indicado.

Dependiendo del valor que se coloque en la posición de memoria 1669, el efecto gráfico se producirá sobre los colores de los registros 0, 2 ó 4 según corresponda.

A modo de ejemplo, entre las líneas 100 y 510 se dibuja una figura en modo gráfico 3, para luego producir la cascada de colores sobre uno de los registros de color escogido.



```

2 REM ** Ideas y Trucos **
4 REM ** ENERO 1985      **
6 REM *****
10 OPEN #1,4,0,"K:"
100 GRAPHICS 3
110 SETCOLOR 4,8,2
120 REM Dibuja Casa
122 COLOR 3:SETCOLOR 2,14,2
130 PLOT 10,7:PLOT 9,8:DRAWTO 11,8
140 PLOT 8,9:DRAWTO 12,9
150 PLOT 7,10:DRAWTO 13,10
160 PLOT 6,11:DRAWTO 14,11
170 PLOT 5,12:DRAWTO 15,12
180 PLOT 7,13:DRAWTO 13,13
190 PLOT 7,14:DRAWTO 13,14
200 PLOT 7,15:DRAWTO 13,15
210 PLOT 7,16:DRAWTO 13,16

```

```

300 REM Dibuja Arbol
302 COLOR 2:SETCOLOR 1,12,2
310 PLOT 30,1:DRAWTO 32,1
320 PLOT 26,2:DRAWTO 36,2
330 PLOT 25,3:DRAWTO 36,3
340 PLOT 24,4:DRAWTO 37,4
350 PLOT 24,5:DRAWTO 37,5
360 PLOT 25,6:DRAWTO 36,6
370 PLOT 26,7:DRAWTO 35,7
380 PLOT 26,8:DRAWTO 35,8
390 PLOT 25,9:DRAWTO 36,9
400 PLOT 24,10:DRAWTO 37,10
410 PLOT 24,11:DRAWTO 37,11
420 PLOT 25,12:DRAWTO 35,12
430 PLOT 26,13:DRAWTO 34,13
432 REM Dibuja Tronco
434 COLOR 3:SETCOLOR 2,14,2
440 PLOT 29,14:DRAWTO 31,14
450 PLOT 29,15:DRAWTO 31,15
460 PLOT 29,16:DRAWTO 31,16
500 REM Dibuja Pasto
502 COLOR 2:SETCOLOR 1,12,4
510 FOR X=1 TO 3:PLOT 0,16+X:DRAWTO 39
,16+X:NEXT X
600 REM Menu de opciones
610 POKE 752,1
612 ? "      1= VERDE      2=CAFE      3=AZU
L"
614 TRAP 610
620 ? :? "          ELIJA UNA OPCION "
:GET #1,NUMERO
630 TRAP 40000:IF NUMERO=49 THEN SW=23
:GOSUB 30000:GOTO 610
640 IF NUMERO=50 THEN SW=24:GOSUB 3000
0:GOTO 610
650 IF NUMERO=51 THEN SW=26:GOSUB 3000
0:GOTO 610
660 IF NUMERO=155 THEN GOTO 610
670 GOTO 610
30000 REM POKE 1669,23=letra
30010 REM POKE 1669,24=fondo
30020 REM POKE 1669,26=marco
30030 REM Cascada
30040 POKE 1664,232:POKE 1665,142:POKE
1666,10
30050 POKE 1667,212:POKE 1668,142:POKE
1669,SW
30060 POKE 1670,208:POKE 1671,234:POKE
1672,206
30070 POKE 1673,254:POKE 1674,6:POKE 1
675,208:POKE 1676,243:POKE 1677,206
30080 POKE 1678,253:POKE 1679,6:POKE 1
680,208:POKE 1681,238:POKE 1682,104:PO
KE 1683,96:POKE 1684,76
30090 POKE 1685,128:POKE 1686,6:POKE 1
790,0:POKE 1789,255
30100 A=USR(1664)
30110 RETURN

```

Digite cuidadosamente el programa y grábalo en cassette o diskette y luego ejecútelo. Una vez dibujada en pantalla la figura, digite el número de la opción deseada y se activará la cascada de colores durante un instante.

# MISCELANEAS

## MANUAL:

# DE RE Atari

## EN ESPAÑOL

El texto más profundo que se ha escrito sobre la programación de los computadores ATARI. A través de sus diez capítulos y diez apéndices los creadores del computador revelan todos sus secretos de hardware y software. Desde una exposición general del sistema hasta los menores detalles de la programación de interrupciones de listas de despliegues para el diseño de gráficas especiales, el uso eficaz del lenguaje ASSEMBLER y del BASIC ATARI para el óptimo aprovechamiento de las virtudes que posee el hardware. Todo está aquí y ahora en castellano, traducido y revisado cuidadosamente, para que el idioma no constituya una barrera para quienes desean o necesitan dominar más y mejor esta extraordinaria máquina que es el computador personal ATARI. Se incorporaron también en la mejor forma posible, sin alterar la línea central del texto las observaciones necesarias relacionadas con las diferencias existentes entre los modelos 400/800 y los XL.

No es en todo caso un libro fácil de leer, ni está pensado para ser una lectura introductoria para el principiante. Requiere concentración y algunos conocimientos previos. Abundan las referencias a los manuales de hardware y del sistema operativo cuyo conocimiento si bien no es indispensable resulta conveniente.

### CONTENIDO

**CAPITULO 1:** Vistazo general al sistema; un corto resumen con un mapa de memoria, un diagrama en bloque y la identificación de las funciones de los tres microprocesadores dedicados: ANTIC, GTIA y POKEY, que además de las PIAS, funcionan bajo la coordinación del microprocesador principal, el 6502.

**CAPITULO 2:** ANTIC y la lista de despliegue, la programación del microprocesador de despliegue con todos los modos gráficos que posee el ATARI y la forma de aprovecharlos en la creación de pantallas combinadas, cambios de página, superposición, etc.

**CAPITULO 3:** Indirección gráfica, registros de color y juegos de caracteres con todo el potencial del manejo de colores para animar las imágenes, la redefinición de caracteres para usos específicos y la creación de gráficas de caracteres.

**CAPITULO 4:** Gráfica de Jugador/Proyectil, una de las facetas más fascinantes y al mismo tiempo menos dominadas de las extraordinarias capacidades gráficas del ATARI. La creación y el manejo de estos simpáticos personajes desde BASIC y desde el lenguaje ASSEMBLER, en aplicaciones de juegos y también otros campos de la programación.

**CAPITULO 5:** Interrupciones de la lista de despliegue; otro potencial muy admirado en sus efectos, pero poco comprendido en su manejo, permite la ejecución de rutinas de máquina en forma sincronizada con la imagen, ya sea para actuar sobre ésta o para otros fines.

**CAPITULO 6:** Movimiento de páginas (scrolling); un poderoso aliado, cuando se trata de exponer al usuario de un programa, una cantidad de información mayor a la que tiene cabida en la pantalla. Textos, diagramas, mapas, etc., pueden desplegarse en todo su tamaño y hacerse visibles, usando la pantalla como una ventana, que puede moverse y apuntarse a voluntad a la región deseada del despliegue. Se tratan los movimientos vertical, horizontal y combinado, tanto grueso como fino.

**CAPITULO 7:** BASIC ATARI, una detallada descripción de las virtudes y de las limitaciones de este BASIC de solamente 8 Kbytes. El funcionamiento del intérprete, uno de los pocos que interpreta línea por línea durante la creación del programa los códigos (token) del BASIC, los punteros, el uso de la memoria y la interacción con el sistema.

**CAPITULO 8:** El Sistema Operativo, una presentación general introductoria, seguida de una descripción más detallada de sub-sistema de entrada/salida (I/O), juego de caracteres en ROM, vectores de rutinas, base de datos, relojes, paquete de coma flotante, etc. y de las posibilidades de su uso explícito desde BASIC y lenguaje de máquina.

**CAPITULO 9:** El Sistema Operativo de DISCO (DOS); la forma en que a través del operador de disco, residente en la ROM del Sistema Operativo, éste se combina con las otras dos partes del DOS: el Administrador de Archivos (FMS) y el Paquete de Utilitarios de Disco (DUP) con detalles de los comandos de disco, estructura de archivos y descripción del procedimiento de acceso aleatorio.

**CAPITULO 10:** Sonido, una completa descripción de los circuitos del ATARI, que le dan una asombrosa capacidad de generación de sonidos. La programación de los poli-contadores y cómo actúan los cuatro canales de audio, incluyendo la combinación de dos de ellos para generar sonidos en un rango de dos bytes (0-65535). La última parte abarca las técnicas de programación para la generación de sonidos.

Los apéndices del 1 al 10 están dedicados al estudio detallado de los siguientes temas: Interrupciones del intervalo vertical, Ingeniería humana. El cassette ATARI. Artificios de Televisión. El paquete de Coma Flotante. Diagrama de flujo del Sistema Central de Entrada/Salida (CIO). El Operador del Impresor Qume. Acceso Aleatorio. Diagrama de Flujo de Inicialización de Encendido. GTIA.

\*\*\*\*\*

Ya se encuentra disponible el volumen I  
del Manual DE RE ATARI en español

\*\*\*\*\*

