# IMPLEMENTACION DE UN MICROMUNDO LOGO TRIDIMENSIONAL A PARTIR DE UNA INVERSION DE LOGO EN DOS DIMENSIONES

RICARDO LUENGO GONZALEZ (+) (*) (&)
MERCEDES MENDOZA GARCIA (&)
TEODORO GONZALEZ BRAVO (+)
LUIS MARQUEZ ZURITA (&)

(+) Departamento de C. Experimentales y Matemáticas de la U. Extremadura.
(*) I.C.E. de la U. Extremadura.
(&) Grupo B.E.T.A.

## INTRODUCCION

Actualmente nadie duda de la utilidad del empleo de los ordenadores en la enseñanza tanto para proporcionar información al alumno como para suministrarle una herramienta con la que pueda desarrollar "ideas poderosas"[1].

Tampoco se duda ya que LOGO es un lenguaje de programación muy adecuado para que el niño apenda a "hablar" con la computadora. Así lo han asumido los responsables de la introducción de la Informática en la Enseñanza, cuando lo adoptan como primer lenguaje de programación[2].

Actualmente se encuentran versiones de LOGO en castellano para casi todos los ordenadores personales (Compatibles MSX, IBM, APPLE, DRAGON, etc). En todos ellos se encuentra implementado el subconjunto gráfico de la Tortuga, micromundo bidimensional que posibilita al alumno tomar contacto con una geometría diferencial distinta de la cartesiana, con la que trabajó hasta ahora, por cierto mucho más "sintónica" son sus intereses propios[3].

Sin embargo vivimos en un mundo tridimensional y sería interesante disponer de una simulación en tres dimensiones manipulable desde LOGO que permitiera al alumno acercarse a la comprensión de muchos fenómenos físicos que ocurren en el ESPACIO-3D.

Ya aparecen en algunos LOGOs primitivas relativas al ESPACIO-3D (como las versions de LOGOSB para MS DOS ya alguna otra) y se encuentran indicaciones en algunos libros de cómo extender el subconjunto de la tortuga al espacio en tres dimensiones, tanto por medio de una proyección paralela[4] como por medio de una proyección cónica. [5] [6].

Los usuarios de ordenadores APPLE disponen de dos versiones de LOGO muy difundidas para el popular APPLE-lle: La suministrada por MICPE S.A.,traducida del inglés al castellano por Paquita Cortada y la versión LOGO II proporcionada directamente por la casa APPLE que exige 128 K de memoria RAM, que puede ser usada tanto en el APPLE IIc como en el IIe con placa de extensión de 64k.

Ninguna de las dos versiones dispone de primitivas para trabajar en 3D y en este artículo tratamos de ofrecer una implementación abierta de un micromundo "3D" construida a partir del micromundo bidimensional para la versión en castellano.

En el apartado 2 exponemos detalladamente el problema de la representación de formas bidimensionales en una pantalla plana centrándonos en el esta-

blecimiento de ejes fijos y móviles y la relación entre ellos por medio de los cosenos directores.

En el apartado 3 definimos los movimientos en el espacio, y calculamos el vector de posición de la tortuga en el estado final, después de haber realizado los movimientos básicos, llegando en cada caso a las ecuaciones de cambio que nos permitiran implementar las primitivas del micromundo.

En el apartado 4 se explicitan las condiciones de contorno y se detallan los procedimientos, haciendo sugerencias de ampliación (que tenemos implementadas, pero que no se incluyen aquí dada la brevedad obligada por las normas de la revista en la que se publica el presente trabajo). Los apartados 5 y 6 proporcionan algunos ejemplos de aplicación del micromundo y las referencias bibliográficas.

## 2. REPRESENTACION DE FORMAS TRIDIMENSIONALES EN UNA PANTALLA PLANA

### 2.1. Proyección cónica de un objeto virtual en tres dimensiones sobre una pantalla plana.

La representación en la pantalla bidimensional de un objeto espacial es similar a la que podemos obtener colocando una cámara fotográfica frente a la pantalla. (Ver figura 1):

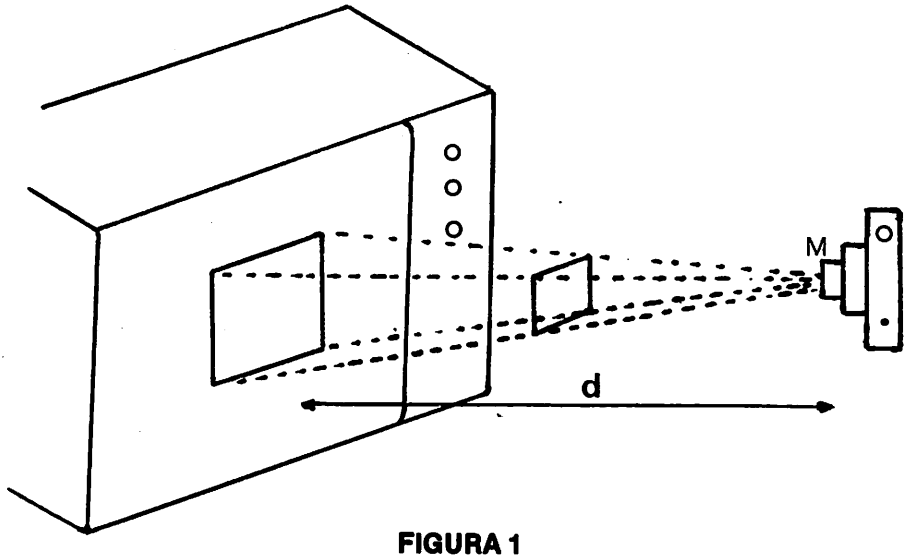



**FIGURA 1**

Esta representación desde una perspectiva *central* (o de proyección cónica) es la que obtendríamos si nos situáramos en un punto de mira M a una distancia de la pantalla *d,* y uniéramos por medio de rayos visuales el punto M con los vértices del contorno del objeto tridimensional, (en este ejemplo un cuadrilátero) y a continuación trazaramos las proyecciones de las aristas del objeto sobre el plano de la pantalla.

Si imaginamos una tortuga tridimensional que fuera capaz de recorrer el objeto en el espacio, (como si el objeto estuviera formado sólo por una estructura hueca, como si sus superficies fueran transparentes, como si observáramos un cuerpo construido con alambre), cada punto de su trayectoria podría unirse con nuestro punto de mira M mediante un "rayo visual".

La intersección de los "rayos visuales" con la pantalla constituye un conjunto de puntos en correspondencia biunivoca con la trayectoria de la tortuga en el espacio virtual 3D. Es la imagen de dos dimensiones que representa la forma supuestamente recorrida por la tortuga tridimensional.

## 2.2. Ejes fijos y ejes móviles propios de la tortuga tridimensional.

Supongamos unos ejes cartesianos ortogonales, cuyo origen sea el centro de la pantalla, orientados como puede observarse en la figura 2:

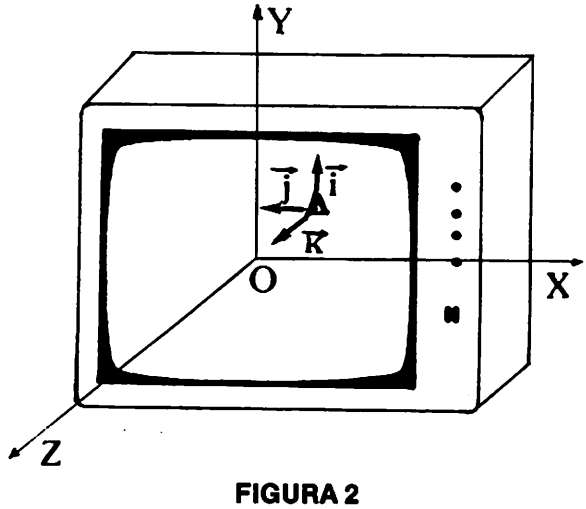



**FIGURA 2**

El eje X es horizontal y (según miramos) su parte positiva se encuentra a la derecha. El eje Y es vertical y su parte positiva se encuentra desde O hacia arriba. El eje Z se encuentra ubicado de tal manera que los tres ejes forman un triedro trirectángulo en orientación positiva (X, Y, Z); es decir que la parte positiva de Z sale hacia fuera perpendicularmente a la superficie de la pantalla.

La tortuga posee unos ejes propios que podemos llamar ($\hat{i}$,$\hat{j}$,$\hat{k}$) situados (ver figura 3):

El eje $\hat{i}$ (definido por este vector unitario) va en el sentido de la marcha de la tortuga. El eje $\hat{j}$ se encuentra situado ortogonalmente al anterior, de forma que la parte positiva se encuentra hacia la izquierda de la tortuga.

El eje $\hat{k}$ se encuentra formando un triedro trirectángulo con los anteriores en orientación positiva ($\hat{i}$,$\hat{j}$,$\hat{k}$) de forma que la parte positiva de este eje sale por encima de la tortuga.

La figura 3 puede representar una tortuga tridimensional. Es un objeto perfectamente orientado, en el que no hay ambiguedad respecto a sus ejes propios, como puede observarse.

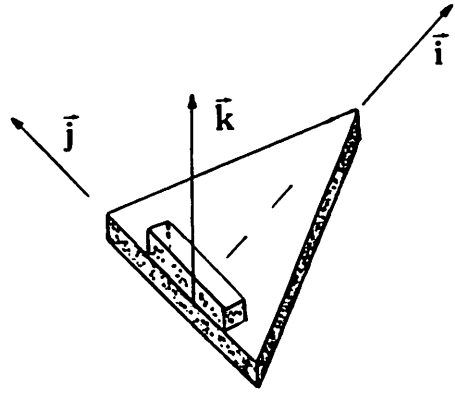



**FIGURA 3**

Es interesante construir esta tortuga, para usarla con los alumnos de EGB, como más adelante indicaremos. Puede servir como modelo un "bloque lógico" de forma triangular al que se puede pegar una pequeña tira rectangular en la parte superior.

Este modelo ha sido empleado por nuestro equipo de trabajo (ref. 7) en el plano bidimensional pero es ideal, a nuestro juicio, para usarlo con los niños en el espacio.

## 2.3. Cosenos directores: relación entre los ejes fijos y los ejes propios.

Tengamos a la tortuga situada en el espacio en un punto cualquiera determinado por su vector de posición $\hat{P}$. (Ver figura 4-a).

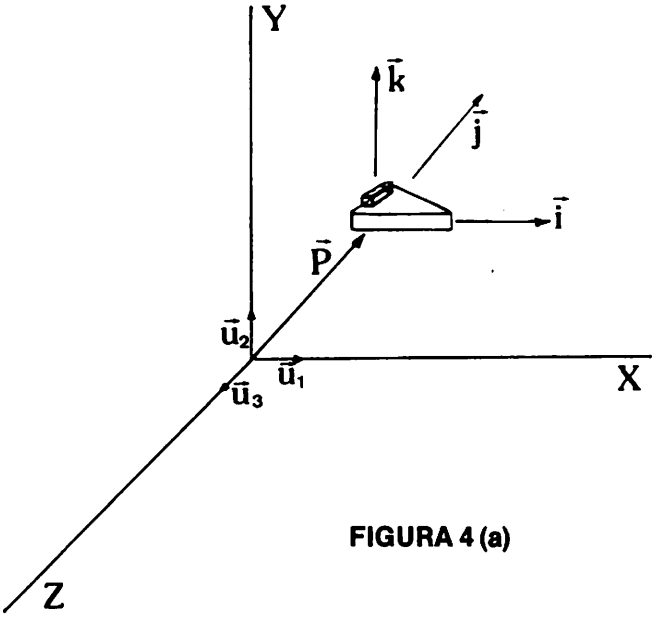



**FIGURA 4 (a)**

Los ejes propios de la tortuga se pueden siempre expresar en función de nuestros ejes fijos. Por ejemplo el vector unitario $\hat{i}$ tomaría la expresión (ver figura 4-b):

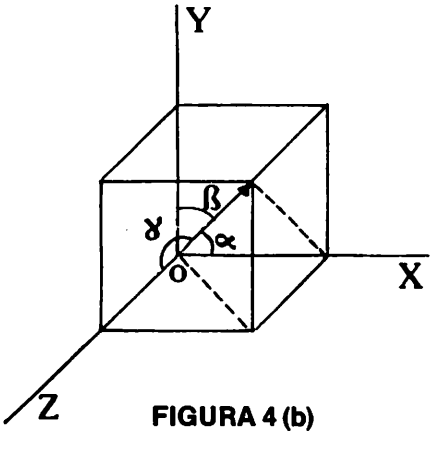



**FIGURA 4 (b)**

(Usualmente llamamos a $\cos\alpha$ , $\cos\beta$ y $\cos\gamma$ , cosenos directores de vector î).

Llamemos a $\cos\alpha = C11$ al objeto de indicar que es el coseno director del primer vector de la tortuga (î) y sobre el primer eje fijo (X).

En general:

Cmn ——→ coseno director del:

n vector unitario de los ejes fijos

m vector unitario de la tortuga

(Variando ambos índices entre 1 y 3).

Por ejemplo: C32 sería el coseno director del vector R de la tortuga respecto del eje fijo Y.

Con esta nomenclatura abreviada, puedo expresar los vectores unitarios propios de la tortuga en función de los fijos como sigue:

$$\vec{i} = C11 * \vec{U1} + C12 * \vec{U2} + C13 * \vec{U3}$$

$$\vec{j} = C21 * \vec{U1} + C22 * \vec{U2} + C23 * \vec{U3} \qquad (I)$$

$$\vec{k} = C31 * \vec{U1} + C32 * \vec{U2} + C33 * \vec{U3}$$

# 3. MOVIMIENTOS EN EL ESPACIO

## 3.1. Desplazamiento de la tortuga en el espacio. Ecuaciones de cambio.

Supongamos la tortuga situada en un punto cualquiera del espacio (1) de coordenadas (x,y,z), cuyo vector de posición es $\vec{P}$. Si efectuamos un desplazamiento hacia el punto (2), dicho desplazamiento viene definido por el vector $\vec{L}$. La nueva posición de la tortuga vendría dada ahora por el vector $\vec{R}$. (Ver figura 5).

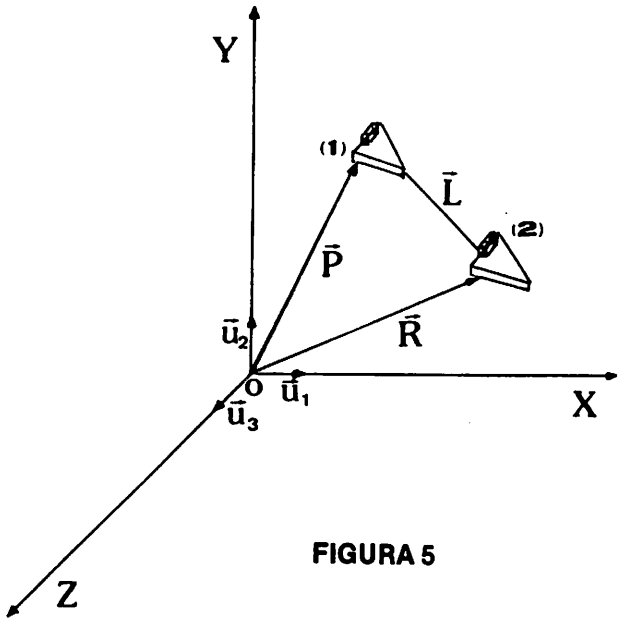



**FIGURA 5**

Es evidente que se cumple la ecuación vectorial:

$$\vec{R} = \vec{P} + \vec{L}$$

Vamos a calcular las componentes del vector $\vec{R}$. Expresemos los tres vectores en función de sus coordenadas cartesianas respecto a los ejes fijos:

$$\vec{P} = x * \vec{U1} + y * \vec{U2} + z * \vec{U3}$$

$$\vec{R} = Rx * \vec{U1} + Ry * \vec{U2} + Rz * \vec{U3}$$

$$\vec{L} = L * \vec{i} = L * (C11 * \vec{U1} + C12 * \vec{U2} + C13 * \vec{U3})$$

(Pues es evidente que la dirección del vector $\vec{L}$ es la misma de la del vector $\vec{i}$ de la tortuga). Hemos llamado L al módulo del vector $\vec{L}$.

Una vez tenemos los vectores expresados en sus componentes respecto de los ejes fijos, sumemos $\vec{P}$ y $\vec{L}$:

$$\vec{P} + \vec{L} = (x + L * C11) * \vec{U1} + (y + L * C12) * \vec{U2} +$$

$$+ (z + L * C13) * \vec{U3}$$

Identifiquemos ahora las componentes de $\hat{P} + \hat{L}$ con las del vector $\hat{R}$:

```
Rx = x + L * C11

Ry = y + L * C12     ( II ) Ecuaciones de cambio
                     ----------------------
Rz = z + L * C13
```

Las nuevas coordenadas del estado (2) después de haber realizado el desplazamiento serán Rx, Ry y Rz calculadas a partir de las ecuaciones de cambio.

Las ecuaciones (II) nos servirán para realizar un procedimiento LOGO que realice el siguiente cometido: Actuando como entrada L (distancia a recorrer), buscará las coordenadas antiguas y los cosenos directores de $\hat{\imath}$, haciendo los cálculos determinados por ( II ), almacenará los nuevos valores y situará la tortuga en la proyección sobre la pantalla real del punto virtual calculado.

Más adelante se detallará el procedimiento LOGO que denominaremos ANDA, que realiza la simulación de un desplazamiento tridimensional.

## 3.2. Proyección de un punto del espacio virtual 3D sobre la pantalla monitora.

Supongamos inicialmente nuestro punto de mira situado a una distancia *d* de la pantalla monitora. (Ver figura 6).

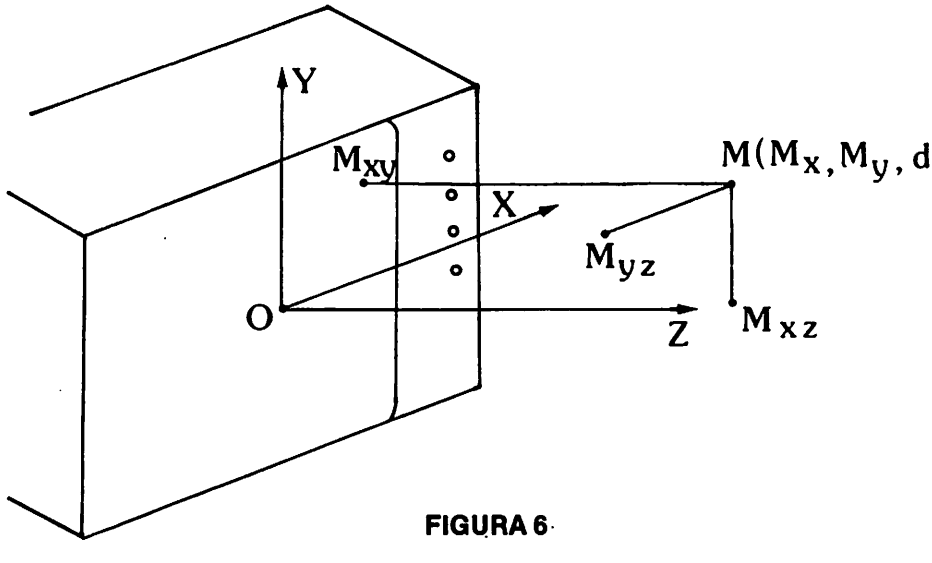



**FIGURA 6**

En general el punto M (de mira) tendrá como coordenadas respecto a los ejes fijos:

$$M\ (Mx, My, d)$$

Imaginemos un punto A (x,y,z) en el espacio que esté situado entre el punto de mira y la pantalla monitora. Calculemos las coordenadas del punto proyección Ap (xp, yp) situado en la pantalla, que es la intersección con esta del rayo visual que parte de M pasando por A.

### Cálculo de la coordenada yp:

Para que resulte menos confusa la figura, vamos a situarnos en la parte negativa del eje fijo X para observar lo que ocurre en el plano (YZ) (ver figura 7):

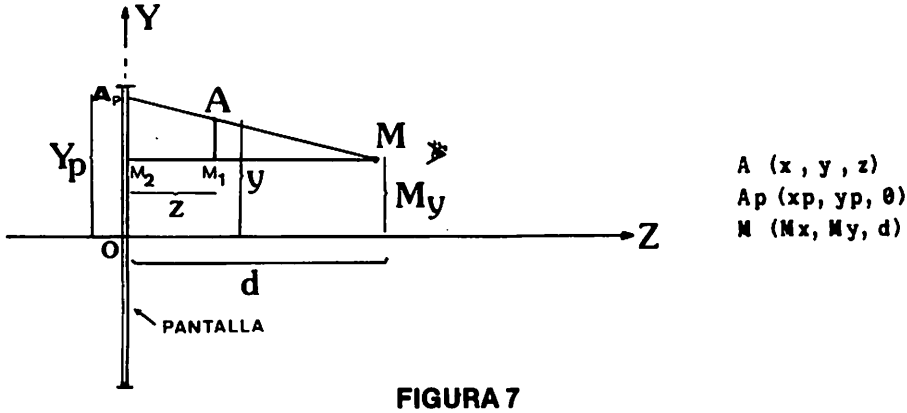



**FIGURA 7**

En la figura 7 hemos tratado de representar el perfil de la pantalla monitora y el eje Z que sale de ella perpendicularmente desde O. Si unimos M con A, la intersección con Y determina el punto Ap (A proyectado).

Tracemos una perpendicular al eje Y desde M, obteniendo el punto M2 en la intersección de ambos. Desde el punto A una paralela al eje Y origina el punto $M_2$ en el corte con la recta anterior $\tilde{M}\,\tilde{M}_2$. Se forman así los triángulos semejantes:

$\widehat{M\,M2\,Ap}$ y $\widehat{M\,M1\,A}$. En ellos se verifica:

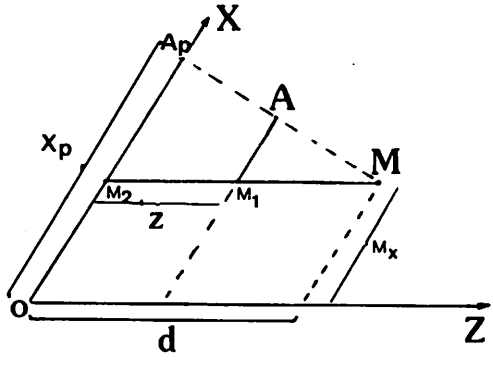
$$\frac{\overline{M\,M2}}{\overline{M\,M1}} = \frac{\overline{M2\,Ap}}{\overline{M1\,A}} \qquad \frac{d}{d-z} = \frac{yp - My}{y - My}$$

En la fórmula son conocidos Mx, My, d, pues son las coordenadas del punto de mira que las define el observador. También conocemos y, z que son dos de las coordenadas actuales del punto A. Podemos entonces despejar yp:

$$yp = My + (y - My)\frac{d}{d - z} \qquad (III)$$

**Cálculo de la coordenada xp:**

Análogamente al caso anterior, vamos a observar ahora qué ocurre en el plano (XZ) (ver figura 8):




A (x , y , z)
Ap (xp, yp, 0)
M (Mx, My, d)

**FIGURA 8**

Centremos nuestra atención ahora en los triángulos semejantes:

$\widehat{M\,M2\,Ap}$ y $\widehat{M\,M1\,A}$ . En ellos se verifica:

$$\frac{\overline{M\,M2}}{\overline{M\,M1}} = \frac{\overline{M2\,Ap}}{\overline{M1\,A}} \qquad \frac{d}{d - z} = \frac{xp - Mx}{x - Mx}$$

Como en el caso anterior conocemos todo menos xp que despejamos:

$$xp = Mx + (x - Mx)\frac{d}{d - z} \qquad (IV)$$

Conocidas las coordenadas xp e yp, podemos realizar un procedimiento LOGO que, como veremos más adelante, vamos a llamar SITUA. Este procedimiento va a situar la tortuga tridimensional en el punto de la pantalla de coordenadas (xp, yp); tendrá como entrada las coordenadas del punto a proyectar (x, y,z) y su efecto será el de realizar la proyección cónica (desde el punto de mira sobre la pantalla) de dicho punto situado virtualmente en el espacio.

### 3.3. Giros de la tortuga tridimensional:

#### 3.3.1. Definición de los tres giros básicos.

Vamos a definir tres giros básicos que llamaremos VIRAR, CABECEAR y BALANCEAR (ver figura 9):

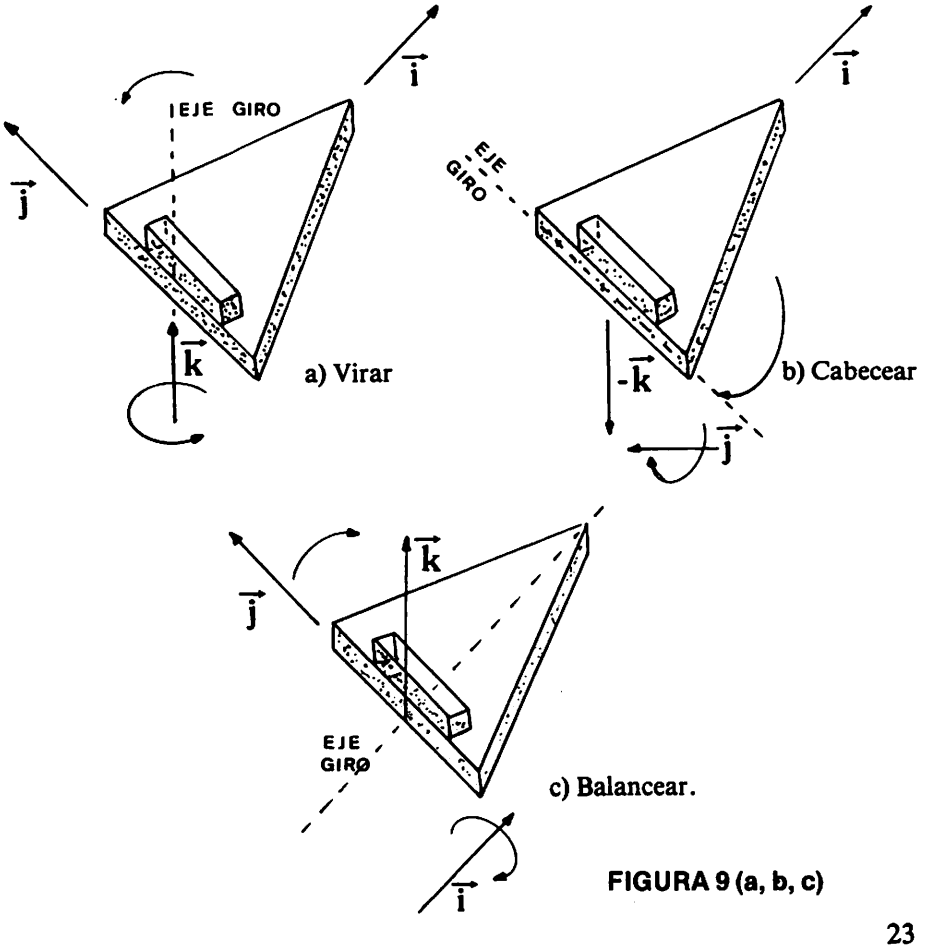



**FIGURA 9 (a, b, c)**

### a) Virar

Es un giro definido por el vector R (en el sentido habitual usado en Física, es decir según la regla del sacacorchos). Este giro es similar al que se produce en el plano y se considera positivo siempre que el eje $\hat{\imath}$ gire sobre $\hat{\jmath}$ en el plano definido por estos dos vectores.

### b) Cabecear

Giro definido por el vector $\hat{\jmath}$. Se considera positivo siempre que $\hat{\imath}$ gire sobre -R en el plano definido por estos dos vectores.

### c) Balancear

Giro definido por el vector $\hat{\imath}$. Se considera positivo siempre que $\hat{\jmath}$ gire sobre R en el plano definido por estos dos vectores.

## 3.3.2. Cálculo de los nuevos cosenos directores después de un giro básico.

Los giros básicos de la tortuga en el espacio siempre se realizan teniendo como eje de giro uno de los ejes propios de la tortuga, por lo que en realidad el giro se realiza en el plano formado por los otros dos ejes.

En general sean dos vectores unitarios ortogonales ($\hat{U}\hat{\imath}$, $\hat{U}\hat{2}$) que definen un plano (ver figura 10):

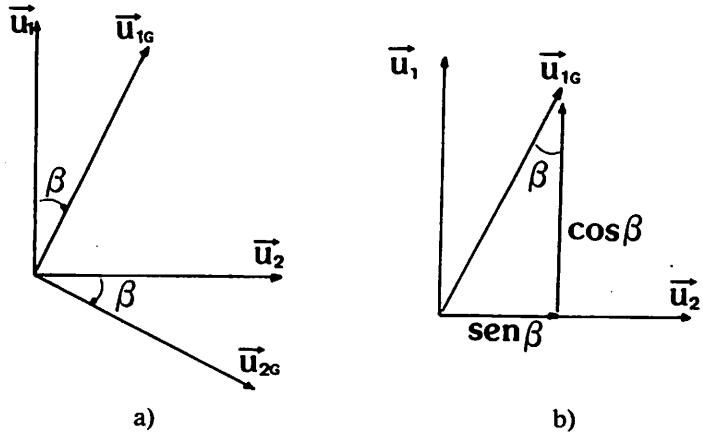



**FIGURA 10 (a y b)**

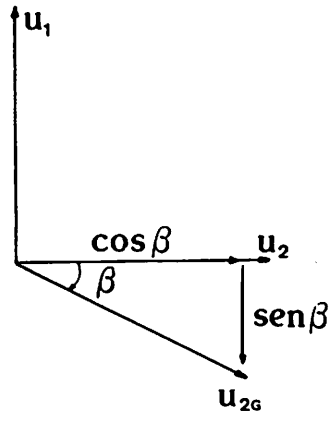



**FIGURA 10 c)**

Consideremos un giro definido por el ángulo $\beta$; calculemos expresiones de los vectores $\hat{U1}\hat{g}$ y $\hat{U2}\hat{g}$, que son los vectores U1 u U2 girados un ángulo

El vector $\hat{U1}\hat{g}$ lo puedo expresar por medio de sus componentes sobre ($\hat{U1}$ $\hat{U2}$) (ver figura 10.b):

$$\vec{U1g} = \vec{U1} * \cos\beta + \vec{U2} * \operatorname{sen}\beta \qquad (V)$$

Procediendo análogamente con $\vec{U2g}$ (ver figura 10.c):

$$U2g = U2 * \cos\beta - U1 * \operatorname{sen}\beta \qquad (VI)$$

Fórmulas que serán de inmediata aplicación a nuestros tres giros básicos.

### 3.3.3. VIRAJE. Ecuaciones de cambio.

En el caso de virar un ángulo $\beta$, las expresiones (V y VI) tomarán ahora la forma: (ver figura 11).

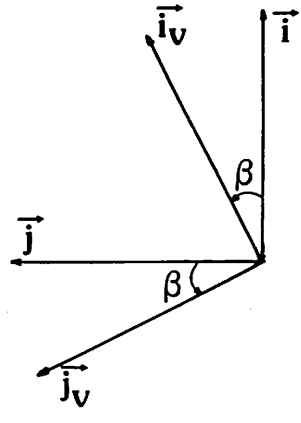



**FIGURA 11**

Llamemos $\hat{i}v$ al eje $\hat{i}$ virado un ángulo $\beta$ y de igual forma $\hat{j}v$ al eje $\hat{j}$ virado ese mismo ángulo (lógicamente pues son ejes ligados).

$$\begin{aligned} \vec{iv} &= \vec{i} * \cos\beta + \vec{j} * \operatorname{sen}\beta \\ \vec{jv} &= \vec{j} * \cos\beta - \vec{i} * \operatorname{sen}\beta \end{aligned} \qquad \text{( VII )}$$

Recordemos las expresiones de $\hat{i}$ y $\hat{j}$ respecto de los ejes fijos (X, Y, Z) en función de sus cosenos directores. (Ver escs. 1).

$$\vec{i} = C11 * \vec{U1} + C12 * \vec{U2} + C13 * \vec{U3}$$

$$\vec{j} = C21 * \vec{U1} + C22 * \vec{U2} + C23 * \vec{U3}$$

Sustituimos estas expresiones en las de los ejes virados ($\hat{i}v$, $\hat{j}v$), (VII) llamando c a $\cos\beta$ y s a $\operatorname{sen}\beta$ :

$$\begin{aligned} \vec{iv} &= c * ( C11 * \vec{U1} + C12 * \vec{U2} + C13 * \vec{U3} ) + \\ &\quad + s * ( C21 * \vec{U1} + C22 * \vec{U2} + C23 * \vec{U3} ) = \\ &= \vec{U1} * (c * C11 + s * C21) + \vec{U2} * (c * C12 + s * C22) + \\ &\quad + \vec{U3} * (c * C13 + s * C23) \end{aligned}$$

Análogamente tendremos para $\hat{j}v$:

$$\begin{aligned} \vec{jv} &= c * ( C21 * \vec{U1} + C22 * \vec{U2} + C23 * \vec{U3} ) - \\ &\quad - s * ( C11 * \vec{U1} + C12 * \vec{U2} + C13 * \vec{U3} ) = \\ &= \vec{U1} * (c * C21 - s * C11) + \vec{U2} * (c * C22 - s * C12) + \\ &\quad + \vec{U3} * (c * C23 - s * C13) \end{aligned}$$

Por tanto, a tenor de las ecuaciones anteriores, cuando viramos un ángulo $\beta$ los cosenos directores de los ejes propios de la tortuga cambian, como se puede ver en la tabla siguiente:

| ejes | antes | despues de virar | siendo |
|---|---|---|---|
| $\vec{i}$ | C11<br>C12<br>C13 | c * C11 + s * C21<br>c * C12 + s * C22<br>c * C13 + s * C23 | |
| $\vec{j}$ | C21<br>C22<br>C23 | c * C21 - s * C11<br>c * C22 - s * C12<br>c * C23 - s * C13 | c = cos β<br>s = sen β |
| $\vec{k}$ | C31<br>C32<br>C33 | C31 No cambian<br>C32 al virar.<br>C33 ------ | |

**TABLA 1**

Conocidos los cosenos directores de la tortuga (Cmn) y el ángulo $\beta$, podemos definir el procedimiento *VIRA*. Tendrá como entrada el ángulo $\beta$, calculará los nuevos cosenos directores con arreglo a la tabla I y actualizará los valores de Cmn antiguos a los nuevos valores calculados.

### 3.3.4. CABECEO. Ecuaciones de cambio.

Las expresiones (V) y (VI) en este caso tomarán la forma: (ver figura 12).

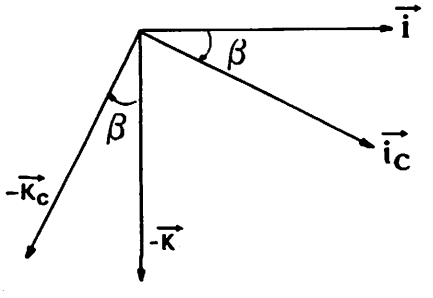



**FIGURA 12**

$$\vec{ic} = \vec{i} * \cos\beta + (-\vec{k}) * \text{sen}\,\beta$$

$$-\vec{kc} = (-\vec{k}) * \cos\beta - \vec{i} * \text{sen}\,\beta$$

Recordemos el valor de $\vec{i}$ y $\vec{k}$ en función de los cosenos directores (fórmulas I):

$$\vec{i} = C11 * \vec{U1} + C12 * \vec{U2} + C13 * \vec{U3}$$

$$\vec{k} = C31 * \vec{U1} + C32 * \vec{U2} + C33 * \vec{U3}$$

Sustituimos los valores de $\hat{\imath}$ y $\hat{R}$ en las expresiones de los ejes virados $\hat{ic}$ y $-\hat{Rc}$, llamando c a $\cos\beta$ y s a $\text{sen}\,\beta$:

$$\vec{ic} = c * (C11 * \vec{U1} + C12 * \vec{U2} + C13 * \vec{U3}) - s * (C31 * \vec{U1} + C32 * \vec{U2} + C33 * \vec{U3}) =$$

$$= \vec{U1} * (c * C11 - s * C31) + \vec{U2} * (c * C12 - s * C32) + \vec{U3} * (c * C13 - s * C33)$$

$$\vec{kc} = c * (C31 * \vec{U1} + C32 * \vec{U2} + C33 * \vec{U3}) + s * (C11 * \vec{U1} + C12 * \vec{U2} + C13 * \vec{U3}) =$$

$$= \vec{U1} * (c * C31 + s * C11) + \vec{U2} * (c * C32 + s * C12) + \vec{U3} * (c * C33 + s * C13)$$

Por tanto cuando cabeceamos un ángulo $\beta$, los cosenos directores de los ejes propios de la tortuga cambian como se puede ver en la tabla siguiente:

| ejes | antes | despues de cabecear | siendo |
|---|---|---|---|
| $\vec{i}$ | C11<br>C12<br>C13 | c * C11 - s * C31<br>c * C12 - s * C32<br>c * C13 - s * C33 | |
| $\vec{j}$ | C21<br>C22<br>C23 | C21 No cambian al<br>C22 cabecear<br>C23 ------- | c = cos β<br>s = sen β |
| $\vec{k}$ | C31<br>C32<br>C33 | c * C31 + s * C11<br>c * C32 + s * C12<br>c * C33 + s * C13 | |

**TABLA II**

A partir de esta tabla podemos definir el procedimiento CABECEA que tendrá como entrada el ángulo $\beta$ y calculará los nuevos cosenos directores de $\hat{\imath}$ y $\hat{k}$ después del cabeceo, actualizando a continuación los valores de las variables que almacenan los cosenos directores.

### 3.3.5. BALANCEO. Ecuaciones de cambio.

Las expresiones (V y VI) tendrán ahora la forma: (ver figura 13):

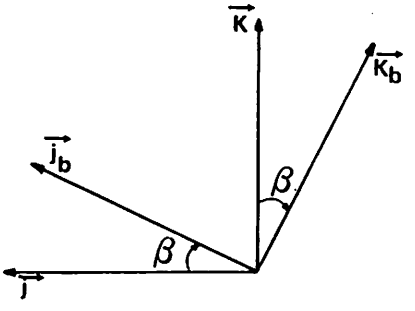



**FIGURA 13**

$$\vec{jb} = \vec{j} * \cos\beta + \vec{k} * \operatorname{sen}\beta$$

$$\vec{kb} = \vec{k} * \cos\beta - \vec{j} * \operatorname{sen}\beta$$

Análogamente a los dos casos anteriores, vamos a utilizar los valores de $\hat{\imath}$ y $\hat{k}$ en función de los cosenos directores (fórmulas I):

$$\vec{j} = C21 * \vec{U1} + C22 * \vec{U2} + C23 * \vec{U3}$$

$$\vec{k} = C31 * \vec{U1} + C32 * \vec{U2} + C33 * \vec{U3}$$

Sustituimos en las expresiones anteriores de $\hat{\jmath}B$ y $\hat{k}B$, llamando c a cos $\beta$ y s a sen $\beta$.

$$\vec{jb} = c * ( C21 * \vec{U1} + C22 * \vec{U2} + C23 * \vec{U3} ) + s * ( C31 * \vec{U1} + C32 * \vec{U2} + C33 * \vec{U3} ) =$$

$$= \vec{U1} * (c * C21 + s * C31) + \vec{U2} * (c * C22 + s * C32) + \vec{U3} * (c * C23 + s * C33)$$

$$\vec{kb} = c * ( C31 * \vec{U1} + C32 * \vec{U2} + C33 * \vec{U3} ) - s * ( C21 * \vec{U1} + C22 * \vec{U2} + C23 * \vec{U3} ) =$$

$$= \vec{U1} * (c * C31 - s * C21) + \vec{U2} * (c * C32 - s * C22) + \vec{U3} * (c * C33 - s * C23)$$

Por tanto cuando balanceamos un ángulo $\beta$, los cosenos directores de los ejes propios de la tortuga cambian como se muestra en la tabla siguiente:

| ejes | antes | despues de balancear | siendo |
|---|---|---|---|
| $\vec{i}$ | C11<br>C12<br>C13 | C11<br>C12<br>C13 No cambian al balancear | |
| $\vec{j}$ | C21<br>C22<br>C23 | c * C21 + s * C31<br>c * C22 + s * C32<br>c * C23 + s * C33 | c = cos $\beta$<br>s = sen $\beta$ |
| $\vec{k}$ | C31<br>C32<br>C33 | c * C31 - s * C21<br>c * C32 - s * C22<br>c * C33 - s * C23 | |

**TABLA III**

Y, de nuevo análogamente a los casos anteriores, podemos definir el procedimiento BALANCEA que tendrá como entrada el ángulo $\beta$, calculará los nuevos cosenos directores de $\hat{j}$ y $\hat{R}$, después del balanceo y actualizará los valores de las variables que almacenan los cosenos directores de los ejes propios de la tortuga ($\hat{i}$ $\hat{j}$ $\hat{R}$) respecto de nuestros ejes fijos (X, Y, Z).

## 4. MICROMUNDO "ESPACIO 3D"

### 4.1. Condiciones de contorno.

Utilizamos para la implementación de este micro- mundo un ordenador Apple IIe de 64 K de memoria RAM con una unidad de disquetes. La versión de LOGO utilizada es la traducción del Apple-Logo en inglés realizada por Paquita Cortada y suministrada por MICPE. S.A. 1983.

El conjunto de procedimientos implementados es una ampliación del conjunto de primitivas, relativas al espacio de tres dimensiones, de las que no dispone esta versión de Logo. Se utilizan para todo el conjunto de procedimientos órdenes básicas del propio Logo, por lo que, a veces, resultan las operaciones de una cierta lentitud y además el manejo del micromundo ocupa mucha memoria.

Cuando cargamos el sistema observamos que quedan 2810 nodos libres, sin cargar el apéndice adicional de utilidades que se suministra en el mismo disquette. Es decir que el total de memoria RAM ya solo quedan (2810 x 5 Bytes aprox. 14 K) después de cargar el intérprete Logo.

La carga del micromundo consume 1695 nodos, por lo que una vez cargado quedan solamente 1115 nodos, es decir 5.5 k libres para el usuario.

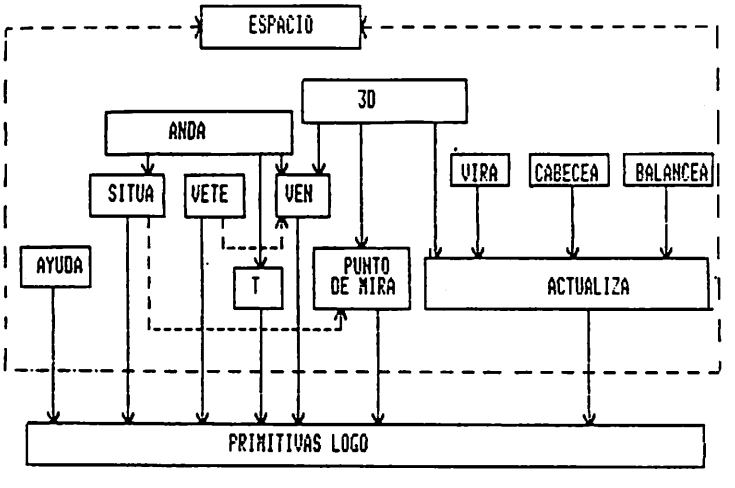
Esta importante limitación debida al consumo de memoria, se ha intentado solucionar a base de incluir la orden RECICLA en un primer procedimiento que llamamos ESPACIO, que debe ser tecleado para activar el micromundo.

Después de teclear ESPACIO pasamos de los 5,5 K a tener ya algo más de 7k.

Los programas recursivos funcionan bien debido a que cuando se agota la memoria el sistema hace un reciclado automático. El problema verdaderamente aparece cuando vamos acumulando procedimientos en el espacio de trabajo. En este caso REPASA seguido de RECICLA puede recuperar algunos nodos más, pero es momento de ir pensando en grabar en disco los programas fraccionadamente para ser borrados de memoria (con lo que ampliamos el espacio de trabajo) y ser llamados solamente aquellos que hagan falta para una tarea puntual.

## 4.2. Procedimientos implementados.

El micromundo programado consta de un conjunto de procedimientos-LOGO (ver listado en páginas siguientes) cuyo diagrama de árbol representamos en el esquema 1.[8].




**ESQUEMA 1**

El procedimiento ESPACIO tiene como misión conjuntar (empaquetar) todos los procedimientos que constituyen el micromundo, para a continuación "enterrar" el conjunto (dejarlo fuera del espacio de trabajo accesible al usuario). Una vez enterrados los procedimientos el programa procede a su autoborrado, con objeto de dejar más memoria libre.

El procedimiento 3D prepara al sistema para actuar al modo Tridimensional estableciendo condiciones iniciales, efectuando un primer borrado y haciendo aparecer la tortuga. Además se ha establecido la escala con unidades iguales sobre los tres ejes y se incluye el modo VENTANA, a fin de que si nos salimos del marco de la pantalla, no se produzcan los arrollamientos típicos del modo LIMITA que harían desaparecer el efecto tridimensional.

La tortuga tridimensional está representada por un pequeño rombo y señala únicamente el lugar en que se encuentra, pero a simple vista no nos proporciona información sobre su orientación. Existen soluciones diversas a este problema (ref. 9) pero todas ellas consumen una porción de memoria que no podemos utilizar, por la escasez que tenemos en la versión LOGO empleada. De todas formas otros intérpretes LOGO con bastante más memoria que el nuestro representan de igual modo la tortuga, teniendo la posibilidad de acceder a los cosenos directores para saber la orientación de la tortuga.

El conjunto de procedimientos VIRA, CABECEA, BALANCEA permiten efectuar los tres giros básicos espaciales de la tortuga definidos en el apartado 3.3.1.(10)

Cada uno de ellos calcula los nuevos cosenos directores, una vez efectuado el giro básico, y actualiza los valores de las variables globales que los almacenan con lo que la tortuga queda enfocada en la nueva dirección y sentido correspondiente a su "estado" final.

El procedimiento ANDA, junto con los anteriores permite efectuar cualquier movimiento en el espacio tridimensional. Si la pluma de la tortuga está baja el "rastro" que deja es verde (para indicarnos que está en el espacio) frente al color de trazo blanco usualmente empleado en el micromundo bidimensional. Las órdenes de alzado y bajada de la pluma son las mismas que las de la tortuga bidimensional. ANDA se sirve de los procedimientos SITUA, VEN y T.

SITUA admite como entradas las coordenadas espaciales de un punto, calcula la proyección cónica de éste teniendo en cuenta la situación del punto de mira (establecida en el procedimiento 3D) y sitúa a la tortuga en el punto de proyección calculado sobre la pantalla.

VEN hace aparecer la tortuga tridimensional, que es dibujada por el procedimiento T. (Si deseáramos dar otra forma a la tortuga no tendríamos más

que modificar este procedimiento). VETE hace desaparecer la tortuga. Ambas pueden ser tecleadas en cualquier momento a modo directo o programado.

ACTUALIZA actúa directamente sobre las variables Cij que almacenan los valores de los cosenos directores. Es el procedimiento que va a efectuar el cambio de los valores de los cosenos directores después de haber efectuado un giro básico o de haber inicializado el sistema.

PUNTO. DE. MIRA establece el "punto de mira", es decir el punto que nos va a servir de origen para realizar la proyección cónica. Tiene como entradas las tres coordenadas espaciales de dicho punto y define las variables que van a almacenarlas. Puede ser llamado tanto a modo directo como en el modo programado y la permanencia del punto se mantendrá salvo en los casos de inicializar el sistema con 3D, o de establecimiento de un nuevo punto de mira por el usuario, con esta misma orden. Es muy útil para desplazarnos alrededor de un objeto y para obtener efectos fotográficos de "gran angular" y "teleobjetivo".

Completa el conjunto el procedimiento AYUDA, independiente de los demás que puede ser llamado en cualquier momento para obtener información acerca de los comandos básicos del micromundo.

A continuación mostramos un listado de los títulos de los procedimientos que constituyen al "micromundo" seguido de los procedimientos completos implementados.

```
PR SITUA :X :Y :Z
PR ACTUALIZA
PR 3D
PR VEN
PR VETE
PR ANDA :L
PR VIRA :B
PR CABECEA :B
PR BALANCEA :B
PR T
PR AYUDA
PR PUNTO.DE.MIRA :M1 :M2 :M3
PR ESPACIO
PR AUX
PR SITUA :X :Y :Z
PONPOS OR ( :MX + ( ( :X - :MX ) * :D ) / ( :D - :Z ) )
( :MY + ( ( :Y - :MY ) * :D ) / ( :D - :Z ) )
FIN

PR ACTUALIZA
HAZ "C11 :T11 HAZ "C12 :T12 HAZ "C13 :T13
HAZ "C21 :T21 HAZ "C22 :T22 HAZ "C23 :T23
HAZ "C31 :T31 HAZ "C32 :T32 HAZ "C33 :T33
FIN
```

```
PR 3D
OT PONRELACION 1
BG BORRATEXTO
VENTANA PONCF 0
CP PONPC 2
PUNTO.DE.MIRA 200 200 600
HAZ "T11 0 HAZ "T12 1 HAZ "T13 0
HAZ "T21 -1 HAZ "T22 0 HAZ "T23 0
HAZ "T31 0 HAZ "T32 0 HAZ "T33 1
ACTUALIZA
HAZ "X 0 HAZ "Y 0 HAZ "Z 0
VEN
FIN

PR VEN
LOCAL "P HAZ "P PLUMA
HAZ "VEN "CIERTO
CP T CUMPLE MU :P
FIN

PR VETE
LOCAL "P HAZ "P PLUMA
PB T CUMPLE MU :P
HAZ "VEN "FALSO
FIN

PR ANDA :L
LOCAL "P HAZ "P PLUMA
SI :VEN = "CIERTO [PB T CUMPLE MU :P]
HAZ "X (:X + :L * :C11)
HAZ "Y (:Y + :L * :C12)
HAZ "Z (:Z + :L * :C13)
SITUA :X :Y :Z
SI :VEN = "CIERTO [CP T CUMPLE MU :P]
FIN

PR VIRA :B
(LOCAL "C "S)
HAZ "C COS :B
HAZ "S SEN :B
HAZ "T11 (:C11 * :C + :C21 * :S)
HAZ "T12 (:C12 * :C + :C22 * :S)
HAZ "T13 (:C13 * :C + :C23 * :S)
HAZ "T21 (:C21 * :C - :C11 * :S)
HAZ "T22 (:C22 * :C - :C12 * :S)
HAZ "T23 (:C23 * :C - :C13 * :S)
ACTUALIZA
FIN

PR CABECEA :B
(LOCAL "C "S)
HAZ "C COS :B HAZ "S SEN :B
HAZ "T11 (:C11 * :C - :C31 * :S)
HAZ "T12 (:C12 * :C - :C32 * :S)
HAZ "T13 (:C13 * :C - :C33 * :S)
HAZ "T31 (:C31 * :C + :C11 * :S)
HAZ "T32 (:C32 * :C + :C12 * :S)
HAZ "T33 (:C33 * :C + :C13 * :S)
ACTUALIZA
FIN

PR BALANCEA :B
(LOCAL "C "S)
HAZ "C COS :B HAZ "S SEN :B
HAZ "T21 (:C21 * :C + :C31 * :S)
HAZ "T22 (:C22 * :C + :C32 * :S)
HAZ "T23 (:C23 * :C + :C33 * :S)
HAZ "T31 (:C31 * :C - :C21 * :S)
HAZ "T32 (:C32 * :C - :C22 * :S)
HAZ "T33 (:C33 * :C - :C23 * :S)
ACTUALIZA
FIN
```

```
PR T
LOCAL "CP
HAZ "CP PC
DE 45
PONPC 4
REPITE 4 [AD 3 IZ 90]
IZ 45
PONPC :CP
FIN

PR AYUDA
MODOTEXTO BORRATEXTO
ES [PRIMITIVAS 'ESPACIO.3D.RL']
ES [_____]
ES []
ES [* 3D - - - > BORRA TODO,PONE MIRA, INI-CIALIZA VARIABLES Y VISUALIZA TORTUGA
ES [] ES [* VEN - - - > HACE VISIBLE TORTUGA]
ES [] ES [* VETE - - - > HACE INVISIBLE LA TORTU-GA]
ES [] ES [* SITUA - - - > SITUA LA TORTUGA EN EL] ES [PUNTO VIRTUAL ( X,Y,Z )]
ES [] ES [* PUNTO.DE.MIRA - - - > ESCOGE EL PUNTO DE MIRA ( MX,MY,D )]
ES [] ES [* ANDA - - - > ANDA TANTOS 'PASOS' COMO INDICA SU ENTRADA]
ES [] ES [* VIRA,CABECEA,BALANCEA - - - >]
ES [GIROS ESPACIALES DE EJES MOVILES RESPECTIVAMENTE K,J,I]
FIN

   PR PUNTO.DE.MIRA :M1 :M2 :M3
   HAZ "MX :M1
   HAZ "MY :M2
   HAZ "D :M3
   FIN

PR ESPACIO
CONJUNTA "ESPACIO [AUX SITUA ACTUALIZA 3D VEN VETE ANDA VIRA CABECEA BALANCEA
T PUNTO.DE.MIRA AYUDA]
ENTERRAR "ESPACIO, BO "ESPACIO RECICLA"
  FIN


  PR AUX
  BORRATEXTO MODOTEXTO
  ES [ESTE PROGRAMA CARGA LOS SIGUIENTES PRO_CEDIMIENTOS AUXILIARES:]
  ES []
  ES [EJES.3D CASA.3D PONX.3D PONY.3D PONZ.3D PONPOS.3D POS.3D]
  ES [TECLEE EL NOMBRE SEGUIDO DE RETURN]
  HAZ "PRO LL
  CARGA PRIMERO :PRO
  ES [DESEAS CARGAR OTRO? ( S / N )]
  HAZ "RES LC
  SI :RES = "S [AUX]
  FIN
```

## 4.3. Sugerencias de ampliación.

Sería interesante disponer de procedimientos auxiliares homólogos a otros frecuentemente utilizados en el micromundo de la tortuga bidimensional que facilitarían la programación en tridimensional.

Debido a la insuficiente memoria disponible no se deben conjuntar con el sistema implementado.

Se pueden almacenar en diskettes y mediante un procedimiento auxiliar, que puede ser añadido al conjunto ESPACIO, se cargan en memoria RAM uno a uno. Una vez utilizados se pueden borrar con objeto de no ocupar "espacio de trabajo".

En caso de utilizar esta ampliación, hay que tener en cuenta que la carga de un programa borra la pantalla de gráficos.

Por ello se deben cargar al comenzar el dibujo los que previsiblemente se vayan a necesitar.

En principio se sugiere la implementación de los siguientes procedimientos:

EJES. 3D, CASA.3D, PONX.3D, PONY.3D, PONZ.3D, PONPOS.3D, POS.3D

Veamos la función que podría realizar cada uno de ellos:

* Ejes.3D: Dibujaría los ejes fijos tridimensionales centrados en la pantalla monitora.

* CASA. 3D: Situaría a la tortuga tridimensional en el origen de los ejes fijos. Si se encuentra en un punto distinto del centro y con pluma traza una línea desde el punto en que se encuentre hacia el centro. Si está sin pluma se desplaza el centro sin rayar.

PONX. 3D: Situaría a la tortuga en el punto del espacio de coordenada X que indique la entrada, manteniendo Y y Z en los valores actuales.

PONY. 3D y PONZ. 3D: Análogamente sitúa a la tortuga en el punto del espacio de coordenada Y, Z (respectivamente), manteniendo las otras dos coordenadas en los valores actuales.

PONPOS. 3D: Admitiendo como entrada una lista compuesta por sus tres coordenadas, sitúa a la tortuga en el punto del ESPACIO.3D de coordenadas (X Y Z).

POS.3D: Orden informativa que informaría de la posición actual de la tortuga. Produciría una lista con las coordenadas (X Y Z) respectivamente.

Este conjunto de procedimientos están implementados por nuestro equipo de trabajo, y no se incluyen en este artículo por los límites de extensión impuestos en la revista.

## 5. EJEMPLOS DE APLICACION

Comencemos con la construcción de un cubo tridimensional. La tortuga debe de recorrer todas y cada una de las aristas, al menos una vez.

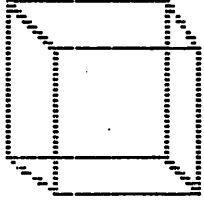


**FIGURA 14**

Planteémonos primero el problema de construir un cuadrado en el espacio. El siguiente procedimiento lo resuelve:

```
PARA CUA :L
REPITE 4 [ANDA :L VIRA -90]
FIN
```

ejecutando: EJES CUA 40

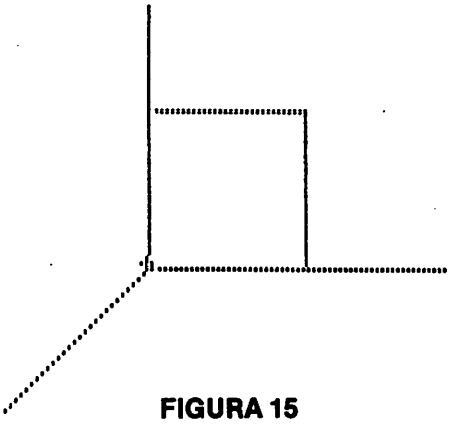


**FIGURA 15**

```
EJES BALANCEA 90 CUA 40
```

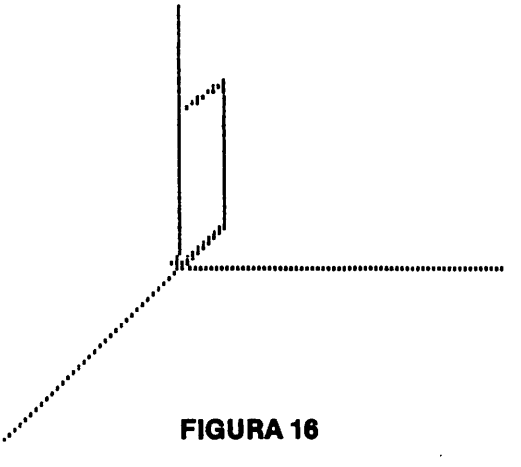


**FIGURA 16**

Trataremos de recorrer el cubo a partir del cuadrado. Una forma de recorrerlo sería:

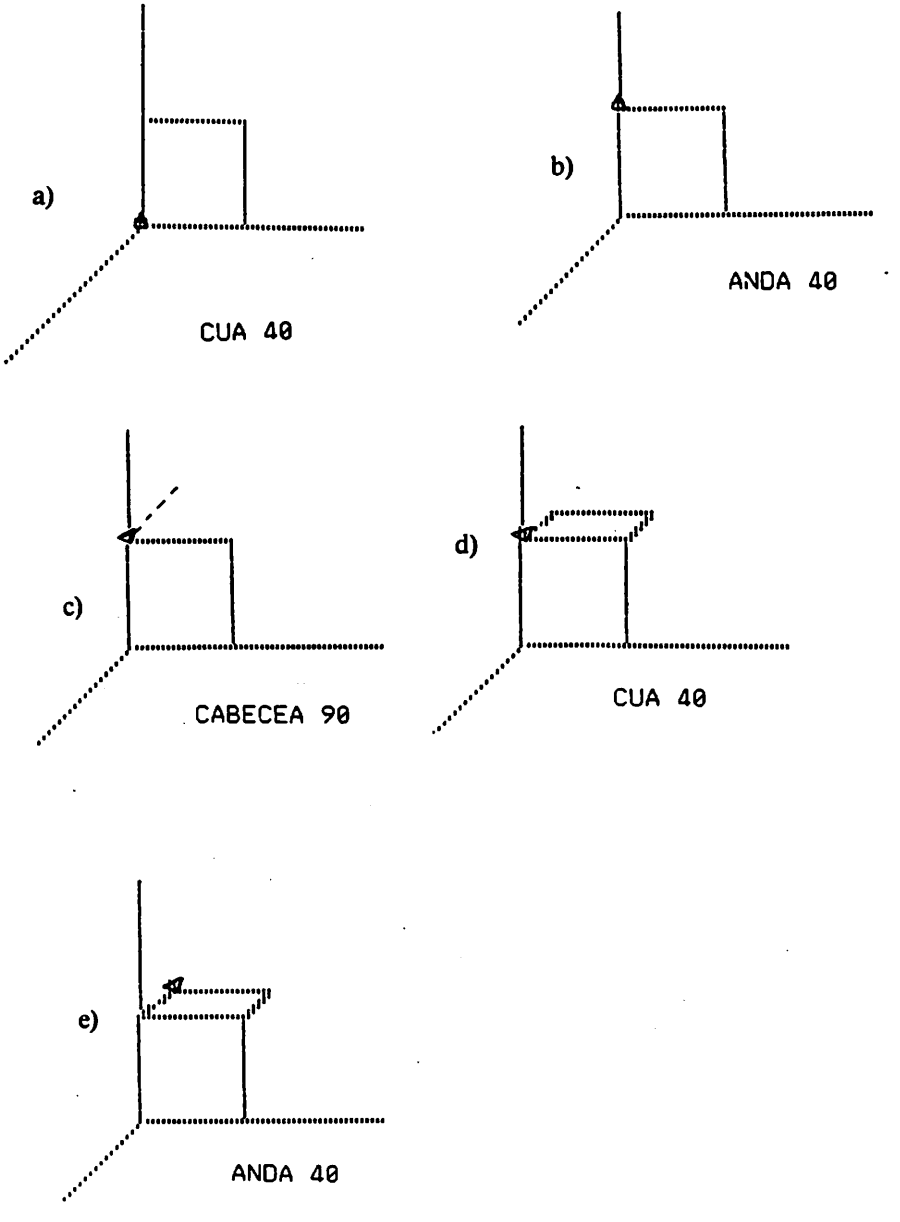



**FIGURA 17 (a, b, c, d, e)**

y el proceso se repetiría cuatro veces. El procedimiento quedaría pues:

```
PARA CUBO :L
REPITE 4 [CUA :L ANDA :L CABECEA 90]
FIN
```

La ejecución CUBO 40 dibujaría definitivamente la figura (14)

En todo ello se ha mantenido el punto de mira en (60 60 600) con objeto de percibir mejor el efecto tridimensional.

La variación del punto de mira nos permite observar el objeto desde distintos puntos de vista. Por ejemplo.

PUNTO. DE. MIRA -200 200 600 CUBO 40

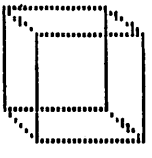


**FIGURA 18**

PUNTO. DE. MIRA -200 -200 600 CUBO 40

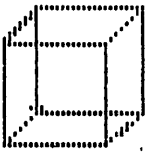


**FIGURA 19**

Una segunda alternativa para observar el cubo tridimensional sería definir un punto de mira fijo sobre el eje Z y colocar el cubo en distintas posiciones.

```
PUNTO DE MIRA 0 0 600
VIRA 30 CABECEA 30 BALANCEA 30          (a)
CUBO 40
FIN
PUNTO DE MIRA 0 0 60
VIRA 30 CABECEA 30 BALANCEA 30          (b)
CUBO 40
FIN
```

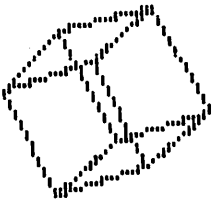
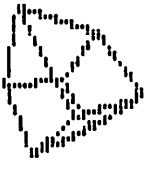
(a)  (b) 

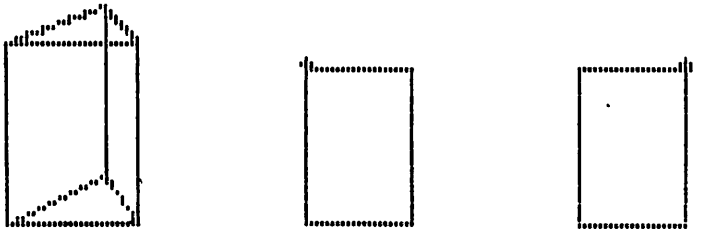
**FIGURA 20 (a, b)**

Vamos a dar otro ejemplo, al menos, de figura tridimensional jugando con ello a modo de "bricolage" para construir modularmente a partir de ella otras figuras tridimensionales.

Construyamos un prisma de bases triangulares. Para ello necesitamos un rectángulo en el espacio:

```
PARA REC :L1 :L2
REPITE 2 ANDA :L1 VIRA -90 ANDA :L2 VIRA -90
FIN
```

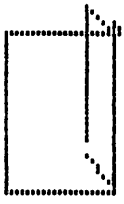
Se trata de construir tres rectángulos ensamblados:



(a) (b) (c)

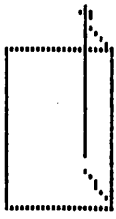
```
VIRA -90 REC 40 60  ANDA 40 CABECEA 120
```

**FIGURA 21 (a, b, c)**



(d)

REC 40 60



(e)

ANDA 40 CABECEA 120

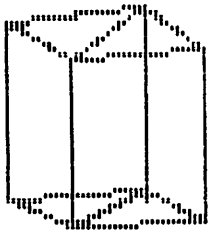
**FIGURA 21 (d, e)**

El procedimiento completo sería:

```
PARA BLOQUE :L1 :L2
VIRA -90
REPITE 3 REC :L1 :L2 ANDA :L1 CABECEA 120
VIRA 90
FIN
```

Dos bloques unidos pueden formar un prisma de bases romboidales:

```
PARA BLTG
BLOQUE 50 70
FIN
BLTG BALANCEA 60 BLTG
```



BALANCEA 15 BLTG BALANCEA 60 BLTG

**FIGURA 22**

Por balanceos sucesivos podemos construir una TARTA:

```
PARA TARTA
BALANCEA 15
PUNTO DE MIRA 200 400 600
REPITE 6 BLTG BALANCEA 60
FIN
```

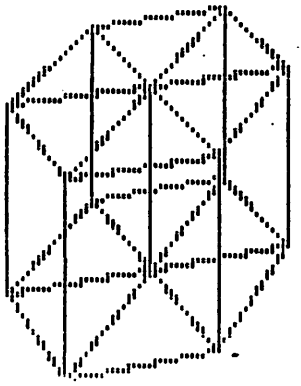


**FIGURA 23**

Separando una porción se consigue aún más ver el efecto tridimensional:

```
PARA TARTAG
SP ANDA 35 CP BALANCEA 30 PUNTO. DE. MIRA 200 400 600
REPITE 5 BLTAG BALANCEA 60
SP BALANCEA 30 VIRA -90
ANDA 55 VIRA 90
BALANCEA -30 CP
BLTG
FIN
```

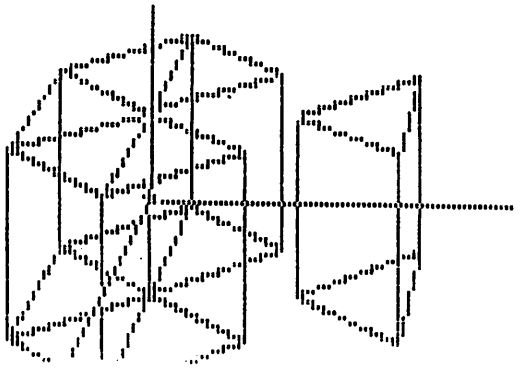


**FIGURA 24**

## REFERENCIAS


1. *"Desafío a la mente"*. S. PAPERT ED. Galápago. Buenos Aires 1984.
   Papert desarrolla en este libro, ya clásico, los principios de la "Filosofía LOGO" y la manera en la que debería utilizarse el ordenador en la enseñanza.
2. *"PROYECTO ATENEA"* MEC. Madrid, 1985.
3. *"Aprendizaje sintónico de la Geometría a través de ordenador: Las primeras proyecciones de los ejes de referencia"*. RICARDO LUENGO y otros. Actas IV Jornada de enseñanza y aprendizaje de las Matemáticas. Tenerife Septiembre de 1984.
4. *"Geometría de tortuga"*. HAROLD ABELSON y A. DI SESSA. ED Anaya Multimedia. Madrid 1986. En este maravilloso libro se pueden encontrar las ideas fundamentales para proceder a implementar un "micromundo-3D" tanto en proyección paralela como en proyección cónica.
5. *"Ideas y Formas"*. HORACIO C. REGGINI. ED. Galápago. Buenos Aires 1985.
   Reggini en este caso efectúa una implementación cónica para TI-PC LOGO en Castellano y proporciona en apéndice las equivalencias del micromundo con la versión inglesa LCSI-LOGO.
6. *"De la tortuga a la inteligencia artificial"* L. RODRIGUEZ ROSELLO. Vector Ediciones. Madrid. 1985.
   Se sugiere en éste una aplicación tridimensional para la versión castellana MS-DOS de IBM PC y compatibles, utilizando también una proyección cónica.
7. *"Proyecto ATREYU"*. RICARDO LUENGO. ANGEL MILLAN. MERCEDES MENDOZA. GENARO MACIAS. MARISA BERMEJO.
   Aprobado por el ICE y el departamento de experimentales y Matemáticas de la UNEX, por convenio con la empresa EUROHARD y con la colaboración del grupo Beta, trata de la aplicación de la geometría logo en un colegio de acción social con niños con graves problemas de adaptación escolar. El modelo aludido ha sido empleado en esta experiencia como "objeto transaccional" en el paso desde una fase "vivenciada corporalmente" hasta una fase de simulación en ordenador.
8. El esquema 1 ha sido dibujado con el programa BEAGLE-GRAPHICS y volacado con TRIPLE DUMP para DOS 3.3. rotando 90.
9. REGGINI en el libro citado en [5] emplea unos ejes destelleantes que pueden ser llamados a voluntad del usuario situados en el punto virtual en que se encuentra la tortuga què proporcionan información de la orientación de la misma, respecto a los ejes fijos.
10. Hemos seguido para la denominación de las nuevas primitivas la "traducción castellana normalizada del LOGO" (LSCI-LOGO para MS-DOS). Boletín de la Asociación LOGO N.° 2 - Septiembre de 1985.