

EL MUNDO DE KAREL: UN POSIBLE PUENTE PARA LA INTRODUCCION AL LENGUAJE LOGO.

RICARDO LUENGO GONZALEZ.

MIGUEL ORREGO CONTRERAS.

TEODORO GONZALEZ BRAVO.

MANUEL BARRANTES LOPEZ..

(Miembros del GRUPO BETA)

1. INTRODUCCION.

El presente trabajo refleja un intento de acercarnos al lenguaje LOGO, a partir de un entorno restringido de programación de apenas media docena de instrucciones.

Basándonos en los estudios de Papert, Reginni y en nuestras experiencias en el campo de la Informática, hemos llegado al convencimiento de que LOGO es el lenguaje actual más adecuado para enseñar a programar en EGB.

Comenzamos en el apartado II, en primer lugar, haciendo unas consideraciones iniciales sobre Informática y Enseñanza, señalando la profusa difusión que se ha alcanzado, así como la falta de modelos metodológicos y criterios contrastados, tanto en la elección del lenguaje como en su enseñanza.

A continuación, se pasa a señalar el reto más importante que hoy día tenemos planteado: la planificación de la enseñanza de la disciplina informática y su utilización como vehículo de transmisión de información.

Cerramos este segundo apartado, centrándonos en el problema de la comunicación del ordenador con el alumno: se analizan las posibles causas de la profusión del lenguaje BASIC y se señala la aparición de lenguajes más adecuados para la EGB, entre ellos LOGO.

En el tercer apartado se fundamenta nuestra opinión que considera a LOGO como un lenguaje de futuro en la Enseñanza y nuestra convicción

de que las ideas fundamentales de la Ciencia pueden ser inducidas en el alumno mediante los recursos que ofrece al ordenador.

Pasamos, en el cuarto apartado, a señalar algunas dificultades que con frecuencia aparecen en el aprendizaje de LOGO y cómo pueden atenuarse si se comienza con un conjunto de actividades previas, realizadas en un entorno de programación, que tiene muy pocas instrucciones y que denominamos "MUNDOKAREL".

Se describe el MUNDO de KAREL, sus movimientos, "las letras" que KAREL puede desarrollar en su mundo y el conjunto de procedimientos que lo implementan a partir de LOGO.

El conjunto de todos los procedimientos que constituyen MUNDOKAREL, no pretende emular a un sistema operativo, puesto que se ha utilizado como base el propio lenguaje LOGO.

Las primeras experiencias realizadas, comenzadas en Enero de 1984 con el proyecto INAM, y los proyectos en curso constituyen el apartado quinto. En éste mostramos también un conjunto de procedimientos para dibujar MUNDOKAREL y varios ejemplos de juegos.

Por último, en el apartado sexto, nuestras conclusiones apuntan hacia la consideración de MUNDOKAREL como un adecuado "puente" hacia el lenguaje LOGO y hacia un cambio en la manera de entender la Educación.

En esencia, hemos pretendido:

- a) Crear, en torno a KAREL, un ámbito de programación que familiarice al alumno con la resolución de problemas sencillos, mediante el ordenador.
- b) Iniciar al alumno en el ámbito de programar de una forma estructurada.
- c) Conseguir que el entorno de programación creado sirva de puente hacia el uso generalizado de LOGO.

2. INFORMÁTICA Y EGB.

2.1. CONSIDERACIONES INICIALES SOBRE INFORMÁTICA Y ENSEÑANZA.

Desde hace unos años y ante el mundo automatizado que se avecina, todos los países se han planteado la introducción de la Informática en los distintos niveles de la Enseñanza.

En España esta preocupación se ha acentuado en la última década, alcanzando la Informática una profusa difusión en todo el Sistema Educativo.

Ha habido un conjunto interesante de experiencias piloto⁽¹⁾, pero, frecuentemente, la implantación de la Informática se ha llevado a cabo "por moda", sin que el Centro Educativo se haya planteado los objetivos concretos y el alcance de la nueva aventura.

En la mayoría de los Centros se ha enseñado a programar (generalmente en BASIC), y se sigue haciendo en la actualidad, sin un método de probada eficacia. La elección del lenguaje se viene haciendo por "invasión comercial" puesto que todos los pequeños ordenadores implementan BASIC.

El método, ha consistido en enseñar a programar a los alumnos lo que se aprendió recientemente en el manual del pequeño ordenador. Los recursos didácticos se basaron únicamente en las intuiciones personales de los profesores sobre cómo se debía enseñar en cada momento.

2.2. EL RETO PLANTEADO EN LA ACTUALIDAD: PLANIFICAR LA ENSEÑANZA DE LA INFORMÁTICA.

Después de la fase, mencionada en el apartado anterior, en la que se produjeron un conjunto importante de tanteos y experiencias aisladas a cargo de improvisados profesores, el reto al que hoy día nos enfrentamos consiste en planificar la introducción de la Informática en nuestro Sistema Educativo.

En nuestra opinión, en esta planificación deberán definirse un conjunto de objetivos y la forma de utilizar la Informática y la tecnología del ordenador en la Enseñanza, en sus dos vertientes:

- Como instrumento de presentar información.
- Como potente herramienta al servicio del alumno.

Todo ello configurado en un Plan Informático Nacional en el que deberán estar previstos:

- Qué contenidos impartir.
- Con qué metodología.
- Cuándo impartirlos (programación temporal de los mismos).
- Con qué medios.
- Plan de Formación Inicial de los Profesores a los distintos niveles.
- Plan de Formación Permanente.
- Plan Transitorio de Implantación Gradual.

Nuestro equipo de trabajo sostiene la necesidad de elaborar ese Plan Informático Nacional, que sirva de referencia para la introducción de la Informática en los Planes de estudio de las Comunidades Autónomas que configuran el Estado Español⁽²⁾.

El problema es complejo y no es nuestra intención, en estas líneas, sino poner de manifiesto la necesidad de abordarlo en profundidad, de una manera coordinada, en todos los niveles de la Enseñanza: desde la propia EGB hasta la enseñanza Universitaria.

2.3. LA COMUNICACION CON EL ORDENADOR: LOS LENGUAJES DE PROGRAMACION.

Nos vamos a centrar en la segunda vertiente aludida en el apartado 2.2.: utilizar al ordenador como herramienta al servicio del alumno.

Uno de los problemas más importantes, para este modo de utilizar el ordenador, es la elección del lenguaje de programación que nos permita comunicarnos con él.

Debemos reconocer, en un principio, que el lenguaje más extendido es el BASIC. En efecto, es el implementado como lenguaje base en la mayoría de los pequeños ordenadores y se utiliza con gran profusión en todos los niveles de la Enseñanza.

BASIC tiene sus partidarios a ultranza y también sus detractores. De BASIC se ha afirmado de todo: desde que es el lenguaje más fácil para un principiante⁽³⁾ hasta que un alumno que programe en BASIC nunca llegará a ser un buen programador; es decir, que BASIC deforma las mentes de los aprendices de programador. En este sentido, Papert⁽⁴⁾ piensa que las razones, que hubo en su día, para la implantación de BASIC han sido superadas en la actualidad.

Si bien hace una década una computadora implementaba BASIC, por la poca memoria necesaria para tal fin, mientras que otros lenguajes potencialmente mejores exigían memorias mayores propias de ordenadores más costosos y voluminosos, hoy día no es ésta una razón válida.

La miniaturización y la incorporación de memorias de gran capacidad en los pequeños ordenadores permiten implementar cualquier otro lenguaje dejando todavía suficiente capacidad libre al usuario.

Sin embargo, Papert sigue opinando⁽⁵⁾ que en nuestra cultura computacional se está detectando un fenómeno de conservadurismo. En concreto, para BASIC se inventan razones "a medida" para justificar su supervivencia, cuando, en realidad, BASIC continúa y continuará, porque:

- Hay muchos ordenadores que implementan BASIC.
- Hay muchos programadores que construyen sus aplicaciones a partir de BASIC.
- Hay muchos programas de aplicación que están contruidos en BASIC.
- Sus pocas primitivas facilitan un rápido aprendizaje inicial.

Las tres primeras razones no tienen nada que ver con la "bondad" del lenguaje y, la última, es una razón de "doble filo" porque, si bien, es fácil aprender esas pocas "primitivas" también es verdad, siempre según Papert, que los programas que hay que realizar para resolver cuestiones no triviales, son difíciles y laberínticos.

En nuestra opinión, aunque en BUP-COU y, por supuesto, en la Universidad, el BASIC puede ser un lenguaje comprensible y utilizable por el alumno no ocurre así en EGB: se está utilizando BASIC para enseñar a programar a los niños, sin pararse a pensar si es el lenguaje más adecuado para este nivel educativo⁽⁶⁾.

La mayoría de los profesores que imparten Informática aplicada a la Educación se han dado cuenta de que hay lenguajes más flexibles, con una mayor versatilidad de uso y una mejor comprensión por parte del alumno de EGB: uno de esos lenguajes es LOGO.

3. LOGO: UN LENGUAJE DE FUTURO EN LA ENSEÑANZA.

En nuestra opinión, muchos de los que piensan que no debería enseñarse a programar en EGB, o bien no conocen LOGO o no conocen la EGB. LOGO más que un lenguaje es una manera de entender la Educación.

En la pg. 17 de su libro "Desafío a la mente", Papert expone su punto de vista sobre la utilización de las computadoras en la Enseñanza: "En muchas escuelas de la actualidad, la frase "instrucción asistida por computadora", significa hacer que la computadora enseñe al niño. Podría decirse que se utiliza la computadora para programar al niño. En mi concepción, el niño programa a la computadora y, al hacerlo, adquiere un sentido de dominio sobre un elemento de la tecnología más moderna y poderosa y a la vez establece un íntimo contacto con algunas de las ideas más profundas de la Ciencia, la Matemática y el arte de construcción de modelos intelectuales"⁽⁷⁾.

El principio en el que se basa LOGO apunta hacia la consideración de que las ideas fundamentales de la Ciencia, las Matemáticas y la construcción de modelos intelectuales que el alumno debe saber, pueden ser inducidas mediante el control, por éste, de los poderosos recursos que ofrece el ordenador.

LOGO es un lenguaje educativo sin umbral y sin techo; es rico en poder de expresión⁽⁸⁾ y presenta ciertas ventajas frente a otros lenguajes dado que:

- a) Hoy día se encuentra traducido íntegramente al castellano.
- b) Lo implementan ya casi todos los microordenadores.

Y en cuanto a sus características como lenguaje:

- c) LOGO es interactivo.

Cualquier orden (primitiva o definida por el usuario) puede ser ejecutada de manera directa sin más que teclearla. la flexibilidad del programa editor LOGO permite fácilmente el diálogo del usuario con el computador a fin de definir, ejecutar y, sobre todo, corregir o modificar sus programas.

d) LOGO es procedimental.

Los procedimientos en LOGO se forman por agrupación de órdenes primitivas y de otros procedimientos anteriormente definidos por el usuario. La llamada a un procedimiento desde otro y así sucesivamente y las comunicaciones entre ellos por medio de las entradas y las salidas pueden llevarnos a niveles de, cada vez mayor, complejidad.

e) El concepto de "objeto LOGO" es de una gran versatilidad y utilidad.

Se corresponde con los datos en otros lenguajes, pero, a diferencia de los mismos, no son objetos LOGO únicamente los números y las cadenas de caracteres, sino también las estructuras compuestas que llamamos listas. El tratamiento de listas es de una gran potencia en LOGO.

f) Incorpora el lenguaje LOGO un subconjunto de órdenes, conocido como "órdenes de gráficos de la tortuga", de especial importancia, para nosotros, en EGB.

LOGO ha demostrado su utilidad en la Escuela Primaria en experiencias realizadas en diversos países y en distintos cursos de la Enseñanza Elemental. Incluso hay experiencias de aplicación a nivel de Preescolar⁹⁾.

A veces se ha confundido LOGO con la "Geometría de la Tortuga" que, si bien, es parte importante de LOGO, no es exclusiva de él¹⁰⁾, ignorando la gran potencia de LOGO en el tratamiento de variables y listas, así como la extraordinaria herramienta de programación que supone la recursión.

En nuestra opinión LOGO no tiene nada que envidiar a otros lenguajes incluso en aplicaciones avanzadas específicas de muchos campos de la Ciencia.

El subconjunto de órdenes de la tortuga es, para nosotros, de particular interés, puesto que la "Geometría de la Tortuga" se apoya en una "Geometría corporal" y el aprendizaje que se produce, en este ámbito de programación, es un aprendizaje sintónico con los intereses del niño¹¹⁾.

El Departamento de Matemáticas de la Escuela Universitaria de Magisterio de Badajoz y el GRUPO BETA¹²⁾ han realizado experiencias de introducción de la Informática en EGB, trabajando con alumnos de Magisterio, Profesores en ejercicio y con los niños de los Centros Públicos de Prácticas Anejos a nuestra Escuela Universitaria¹³⁾.

Los resultados obtenidos en estas primeras experiencias hacen augurar que, en un futuro no muy lejano, se pueda llegar a conclusiones importantes que apunten hacia la utilización de LOGO en EGB, como uno de los lenguajes más adecuados para este nivel educativo¹⁴⁾.

4. UNA RESTRICCIÓN DE LOGO: EL ENTORNO DE PROGRAMACIÓN "MUNDOKAREL"⁽¹⁵⁾.

4.1. DIFICULTADES EN EL APRENDIZAJE DE LOGO.

A pesar de nuestras afirmaciones, del apartado anterior, sobre LOGO, en el sentido de considerarlo uno de los lenguajes actuales de alto nivel más adecuados para la enseñanza y, en concreto, para la EGB, su aprendizaje no está exento de dificultades.

De nuestra experiencia ALCA-84⁽¹⁴⁾ podemos señalar las que aparecieron con más frecuencia en los niños de tercero de EGB:

- Dificultad con el teclado (lentitud).
- Dificultad con el manejo del espacio de trabajo.
- Dificultad con la utilización de variables-LOGO.
- Dificultad en comprender y utilizar la gran potencia de la recursión.
- La presentación, en poco tiempo, de muchas primitivas oscurecía la visión general del lenguaje.

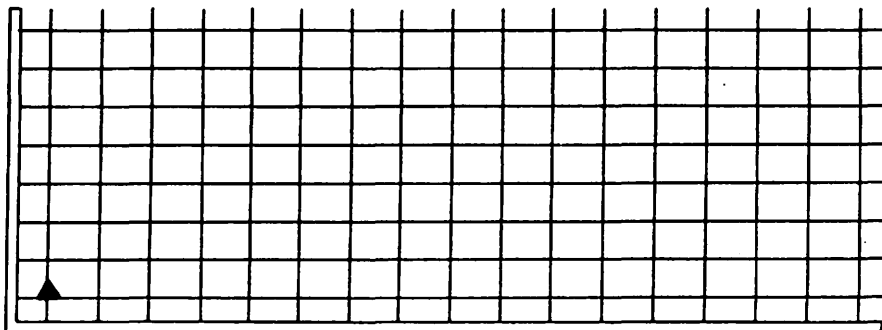
Todas estas dificultades nos obligaron a reflexionar sobre la necesidad de elaborar una Didáctica de la Informática y, en particular, una Didáctica-LOGO.

La última de las dificultades señaladas, en los párrafos precedentes, nos llevó a pensar que podía ser más fácil llegar a aprender LOGO comenzando con una serie de actividades previas, en un entorno de programación que tiene apenas media docena de primitivas, denominado por nosotros "MUNDOKAREL".

4.2. LA RESTRICCIÓN DE LOGO UTILIZADA: "MUNDOKAREL".

KAREL es un autómatas que circula en su mundo atendiendo a un conjunto de instrucciones muy reducido. Se ha utilizado la propia tortuga LOGO para simular a KAREL⁽¹⁶⁾.

El mundo de KAREL que hemos implementado, a partir de LOGO, se compone de una red de 8 vías horizontales y 17 verticales, cortadas perpendicularmente que parten de los "límites", como puede verse en la Figura 1.



— FIGURA 1 —

KAREL dispone de movimientos: avance y giro, que le permiten desplazarse (o girar) pasando de un nudo a otro de la red; además, puede realizar tareas con unos elementos que llamaremos "emisoras".

Estos son objetos cuadrados de pequeño tamaño que KAREL puede coger o soltar en los nudos de la red. Cuando coge un emisor lo mete en su "bolsa", sabiendo los emisores que tiene. Si dispone de uno o más emisores puede dejarlos en los nudos de la red. Al intentar soltar un emisor, cuando KAREL no tiene ninguno, nos informa ¡bolsa vacía! regresando a ALTO NIVEL. KAREL sabe pararse y desconectarse y con esta acción termina normalmente cualquier tarea.

KAREL atiende a cuatro instrucciones básicas:

- 1) SALTO.
- 2) IZQUIERDA.
- 3) COGEREMISOR.
- 4) SOLTAREMISOR.

Tiene, también, la capacidad de aprender instrucciones y ejecutar una tarea mediante el procedimiento que está formado por ese conjunto de órdenes. Basta "ordenarle" el nombre del "procedimiento aprendido" para que ejecute la tarea.

Es interesante la simulación manual que propone Angel Martín⁽¹⁷⁾ consistente en una cuadrícula pintada sobre una cartulina y una tortuga simulada en el tamaño adecuado que el niño puede mover por los nudos de dicha cuadrícula.

Recomendamos que, antes de pasar a la simulación en ordenador, se trabajen varios ejemplos sobre este modelo inicial.

4.3. CONJUNTO DE PROCEDIMIENTOS QUE CONSTITUYEN EL MUNDOKAREL.

Se ha utilizado en la simulación un ordenador APPLE-IIe de 64k con unidad de disquettes. La versión de LOGO utilizada es "LOGO en castellano" suministrado por MICPE S.A. 1983.

El conjunto de procedimientos que simula "MUNDOKAREL" no pretende emular a un sistema operativo, puesto que se ha utilizado como base el propio lenguaje LOGO. Precisamente se utiliza el editor de LOGO para editar los procedimientos de KAREL, lo cual permite al alumno familiarizarse con la manera de construir y modificar los procedimientos en LOGO.

El alumno podrá así pasar de KAREL a LOGO mediante una especie de "ósmosis" puesto que en realidad está utilizando un LOGO restringido a muy pocas instrucciones.

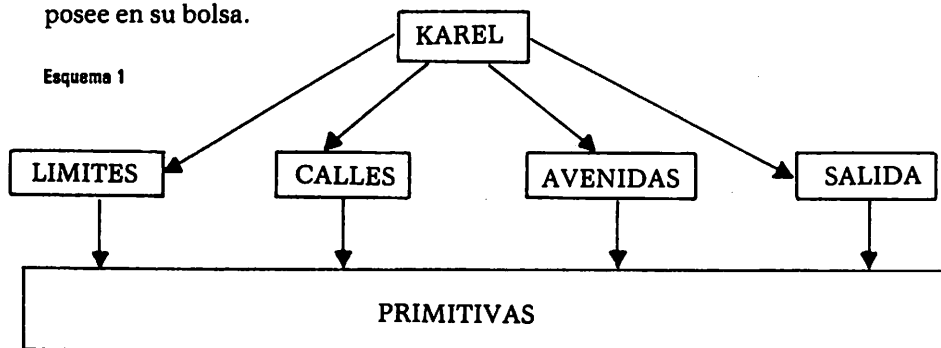
Se observa un inconveniente en cuanto a la lentitud que presenta KAREL y a ciertas paradas de segundos que efectúa el autómat. Las órdenes REPASA y RECICLA solucionan, en parte, el segundo inconveniente aludido liberando algunos NODOS. Se puede dar la circunstancia de que el alumno teclee una orden LOGO que no pertenezca a KAREL y el autómat le obedezca e, incluso, se salga de los nudos de la red. La orden SALIDA puede solucionar este incidente si se desea regresar a los nudos de la red. Pero lo importante es que, en este caso, hemos comprobado que el alumno está ya preparado para "dar el salto" de KAREL a LOGO.

El entorno de KAREL programado consta de dos conjuntos de procedimientos-LOGO cuyos diagramas de árbol representamos en los esquemas 1 y 2:

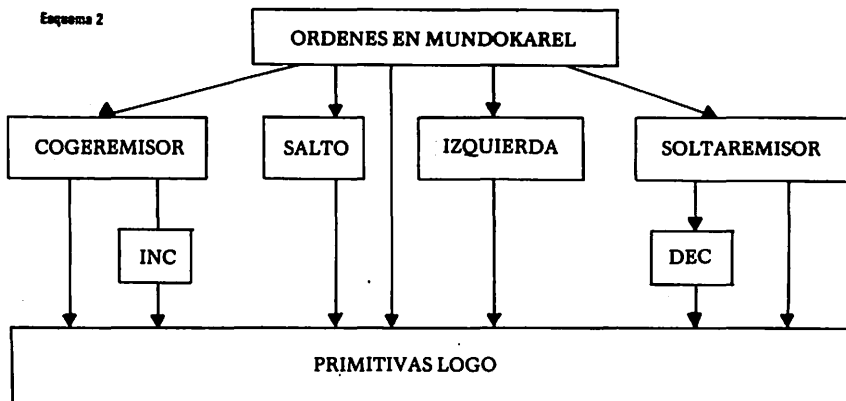
a) El conjunto (KAREL, LIMITES, CALLES, AVENIDAS y SALIDA) que constituye el MUNDOKAREL, ya descrito, sitúa al autómat en el punto inicial de partida (SALIDA).

b) El conjunto de órdenes en MUNDOKAREL (SALTO, IZQUIERDA, SOLTAREMISOR, COGEREMISOR, INCREMENTAR y DECREMENTAR) que constituyen los procedimientos-LOGO simulan las órdenes básicas de KAREL con INC y DEC que permiten saber los emisores que KAREL posee en su bolsa.

Esquema 1



Esquema 2



Una exposición gráfica detallada sobre los procedimientos correspondientes al dibujo en pantalla del mundo de KAREL, se ilustra en la figura 2. (a) (b) (c) (d) (e).

PR LIMITES

OT SP PONPOS [-130 -50]

CP AD 130

DE 90 AD 5

DE 90 AD 125

IZ 90 AD 255

DE 90 AD 5

DE 90 AD 260

DE 90 SP

FIN

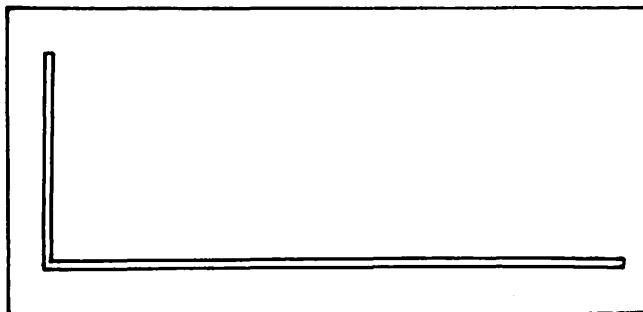


FIGURA 2(a)

PR CALLES

SP PONPOS [-125 -35] CP

REPITE 3 [DE 90 AD 255 SP IZ 90 AD 15 CP IZ 90 AD 255 DE 90 AD 15]

DE 90 AD 255 SP

IZ 90 AD 15 CP

IZ 90 AD 255 DE 90

FIN

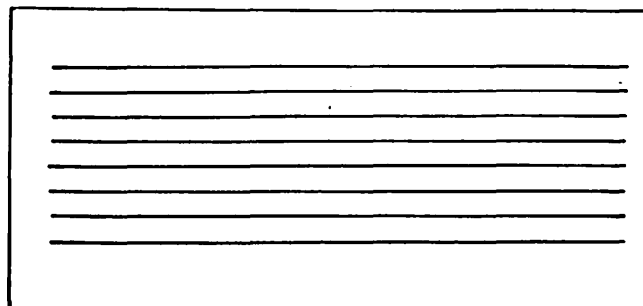


FIGURA 2(b)

**PR AVENIDAS
SP PONPOS [-115 80] CP
REPITE 8 [AT 125 DE 90 AD 15 IZ 90 AD 125 DE 90 SP AD 15 CP IZ 90]
AT 125
FIN**

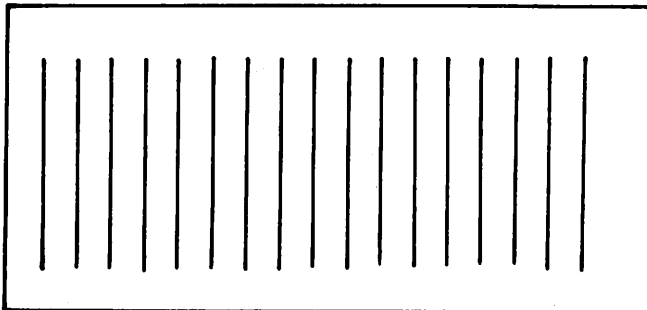


FIGURA 2(c)

**PR SALIDA
SP PONPOS [-115 -35]
PONR 0
HAZ "E O PT
FIN**

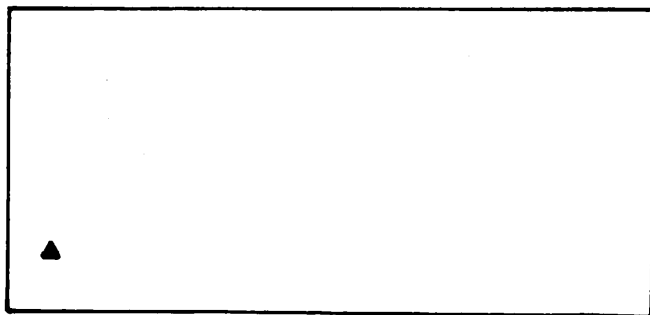


FIGURA 2(d)

**PR KAREL
LIMITES
CALLES
AVENIDAS
SALIDA
FIN**

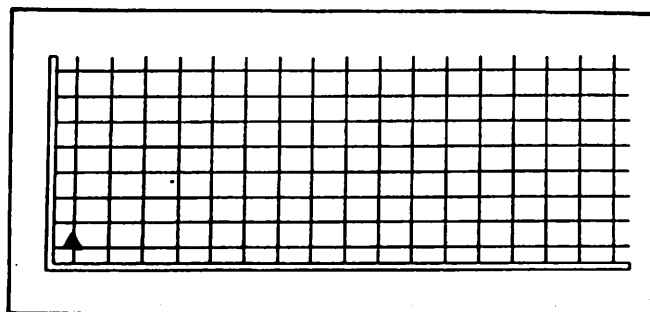


FIGURA 2(e)

En esta figura hemos colocado los procedimientos juntamente con los efectos que producen las ejecuciones de los mismos en el monitor.

El procedimiento LIMITES dibuja los límites inferior e izquierdo de MUNDOKAREL. Los procedimientos CALLES y AVENIDAS permiten trazar la red que constituye el mundo donde circula el autómeta.

Se han preferido las órdenes de movimientos relativas a los ejes propios de KAREL, antes de utilizar órdenes LOGO relativas a coordenadas cartesianas. El motivo principal que nos ha llevado a programarlas así es la posibilidad de que el alumno, una vez en LOGO, pueda editar y comprender estos procedimientos. Se ha ocultado la tortuga a fin de mejorar la velocidad de trazado.

El procedimiento SALIDA sitúa a KAREL en el punto de partida mediante las órdenes LOGO PONPOS (-115, -35) y PONR 0, descargando todos los emisores que tuviera KAREL en su bolsa y haciendo visible la tortuga. Esta orden puede teclearse en cualquier momento. Si la tortuga está oculta, la hace aparecer, pero si está visible no la esconde.

El procedimiento KAREL se compone de los anteriores, basta teclear su nombre (seguido como siempre de RETURN) para que se dibuje, en el monitor, MUNDOKAREL y se sitúe KAREL en el punto de salida.

Vamos a comentar, brevemente, los procedimientos que simulan las órdenes de MUNDOKAREL:

PR COGEREMISOR

AD 2 PB

IZ 90 AD 2

REPITE 3 [IZ 90 AD 4]

IZ 90 AD 2

DE 90 SP

AT 2 INC "E

ESCRIBE OR [BOLSA:] :E

FIN

PR SOLTAREMISOR

TEST :E < 1

SICIERTO [ESCRIBE [BOLSA VACIA]

AD 2 CP

IZ 90 AD 2

REPITE 3 [IZ 90 AD 4]

IZ 90 AD 2

DE 90 SP

AT 2 DEC "E

ESCRIBE OR [BOLSA:] :E

FIN

PR SALTO

SP AD 15

FIN

PR IZQ

IZ 90

FIN

PR DEC :X

SI NO NOMBRE? :X [STOP]

SI NUMERO? COSA :X [HAZ :X SUMA

FIN COSA :X -1]

PR INC :X

SI NO NOMBRE? :X [STOP]

SI NUMERO? COSA :X [HAZ :X 1

FIN + COSA :X]

PR KAREL

LIMITES

CALLES

AVENIDAS

SALIDA

FIN

El procedimiento SALTO hace que la tortuga pase de un nudo de la red al siguiente en la dirección y sentido en que se encuentre orientada: IZQ realiza un giro de 90 grados a la izquierda con objeto de mantener a KAREL siempre en la red ortogonal.

Para realizar un conjunto de tareas con los emisores, que detallaremos más adelante, KAREL atiende a dos órdenes: COGEREMISOR y SOLTAREMISOR. La primera simula la recogida del emisor mediante el borrado del pequeño cuadrado situado en el nudo donde se encuentra la tortuga. El procedimiento elaborado al efecto llama también a otro procedimiento que denominamos INC que incrementa en una unidad la variable "E que almacena el número de emisores que actualmente tiene en su bolsa. Así mismo nos informa de cuál es el número final de emisores."

La segunda orden (SOLTAREMISOR) simula la colocación de un emisor en el nudo donde actualmente se halla KAREL, dibujando un pequeño cuadrado cuyo centro es el propio nudo. Igual que en el caso anterior, el procedimiento llama a otro que denominamos DEC, que disminuye en una unidad la variable "E, ya aludida, y análogamente nos informa del número final de emisores en bolsa."

5. PRIMERAS EXPERIENCIAS REALIZADAS.

En Enero de 1984 se comenzó el proyecto "INAM" (INFORMATIZACION ANEJAS MAGISTERIO), cuyo objetivo principal era establecer un AULA PILOTO INFORMATICA (A.P.I.) común al Departamento de Matemáticas de la E.U.M. de Badajoz y a las Escuelas Anejas, a la vez que introducir a los profesores en la aplicación de la Informática en la Enseñanza.

Gracias a este proyecto tuvimos la oportunidad de desarrollar la experiencia de enseñar LOGO y ensayar MUNDOKAREL con niños de EGB (proyecto "ALCA-84") y con Profesores de EGB en ejercicio (proyecto "ALCA-85").

Asimismo, con nuestros alumnos de la E.U. de Magisterio desarrollamos el proyecto "Informática para profesores de EGB", financiado por la Subdirección General de Investigación Educativa. Con él hemos completado los tres sectores directamente implicados en la introducción de la Informática en EGB.

Fruto de estas experiencias son los procedimientos que mostramos en los siguientes apartados.

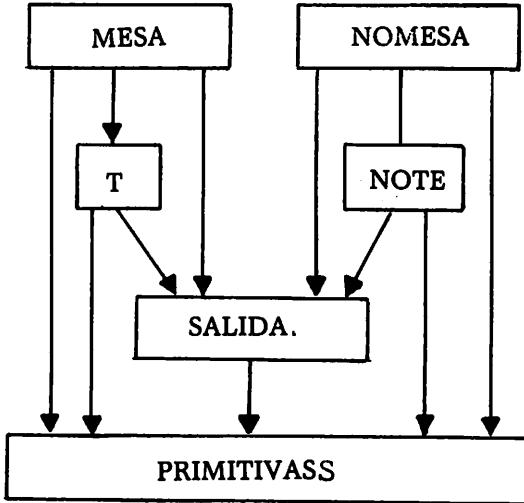
5.1. PROCEDIMIENTOS PARA DIBUJAR EN MUNDOKAREL.

Los procedimientos que siguen han sido realizados por niños de tercero de EGB. El mundo de KAREL presenta la posibilidad de dibujar letras con los emisores, dejándolos en los nudos adecuados de la red. Posteriormente,

esas letras, combinadas consigo mismo o con otras, pueden dar lugar a interesantes dibujos.

En este caso se ha implementado un conjunto de procedimientos cuyas relaciones lógicas quedan representadas en el árbol del esquema 3.

- ESQUEMA 3 -



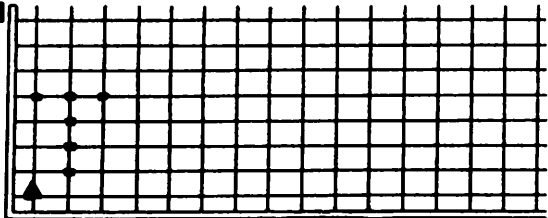
El procedimiento T sugirió la elaboración de NOTE que borra la T dibujada anteriormente. Figura 3.

```

PR T
HAZ *E 8 DER S IZQ
REPITE 3 [S SOLTAREMISOR]
IZQ S DER S DER
REPITE 3 [SOLTAREMISOR S]
SALIDA
FIN
  
```

```

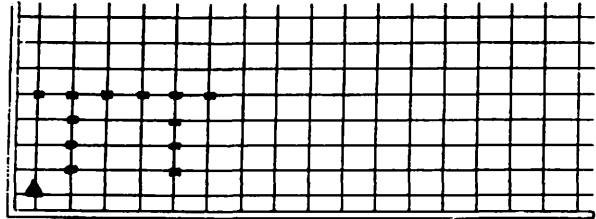
PR NOTE
HAZ *E 6 DER S IZQ
REPITE 3 [S COGEREMISOR]
IZQ S DER S DER
REPITE 3 [COGEREMISOR S]
SALIDA
FIN
  
```



- FIGURA 3

La propia T desplazada y repetida sugiere una mesa. Figura 4.

PR MESA
 SALIDA
 T
 DER S S S
 IZQ
 T
 FIN



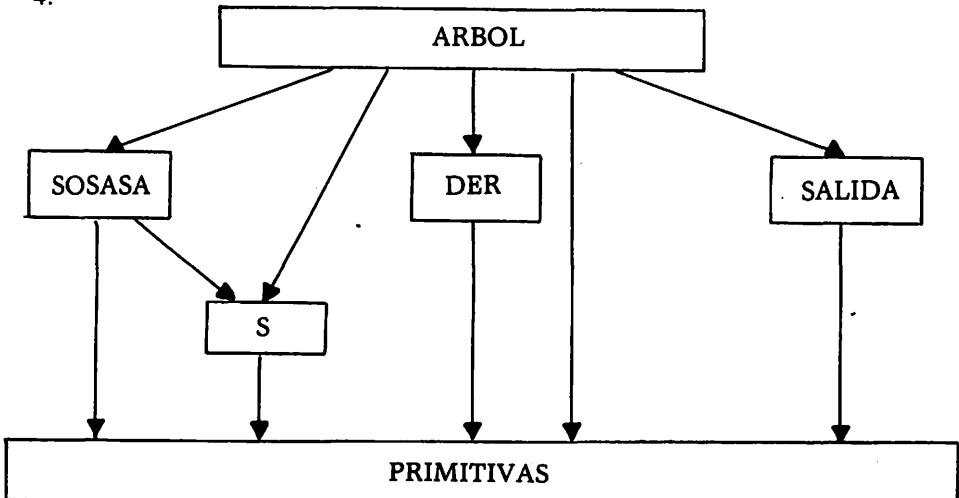
PR NOMESA
 SALIDA
 NOTE
 DER S S S
 IZQ
 NOTE
 FIN

- FIGURA 4

El procedimiento NOMESA borraría, de la misma manera que hicimos con el ejemplo anterior, (NOTE) la mesa dibujada por el procedimiento MESA.

El procedimiento ARBOL utiliza las primitivas de KAREL apoyándose, así mismo, en tres procedimientos implementados a partir de aquellas (SOSASA, S y DER).

El diagrama del procedimiento ARBOL se implementa en el esquema 4.

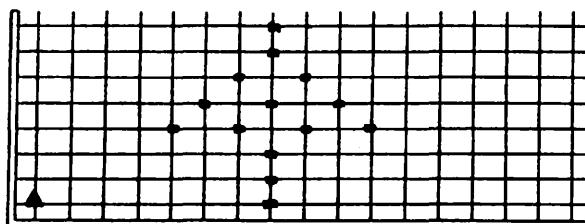


PR ARBOL
HAZ *E 20
DER
REPITE 7 [S]
IZQ
REPITE 3 [SOLTAREMISOR S]
S SOSASA
REPITE 2 [SOLTAREMISOR S]
IZQ IZQ
REPITE 5 [S]
DER S S S
IZQ IZQ
REPITE 4 [SOSASA]
IZQ S IZQ S S S
REPITE 3 [SOSASA]
DER S DER S S S
REPITE 2 [SOSASA]
SALIDA
FIN

PR DER
REPITE 3 [IZQ]
FIN

PR SOSASA
SOLTAREMISOR
S S
FIN

PR S
SALTO
FIN



En el listado de este procedimiento observamos, en primer lugar, la toma de 20 emisores, suficientes para dibujar la correspondiente figura.

"S se trata de una simplificación de la orden "SALTO debida a la gran utilización de esta primitiva de KAREL.

El procedimiento SOSASA efectúa la suelta de un emisor en el lugar en que se encuentra la tortuga, efectuando, a continuación, dos saltos en el sentido de la marcha.

En el dibujo del árbol se va a repetir este proceso un número significativo de veces, suficientes para justificar la implementación del procedimiento.

La orden "DER constituye un procedimiento implementado a partir de la repetición de la primitiva de KAREL IZQ y proporciona una cierta comodidad en el movimiento de KAREL al efectuar el dibujo.

Presentamos el procedimiento tal y como lo construyeron nuestros alumnos, probablemente se podía haber estructurado "ARBOL en "EJE, "COPA y los correspondientes procedimientos "SITUA que ensamblados efectuarían el mismo dibujo.

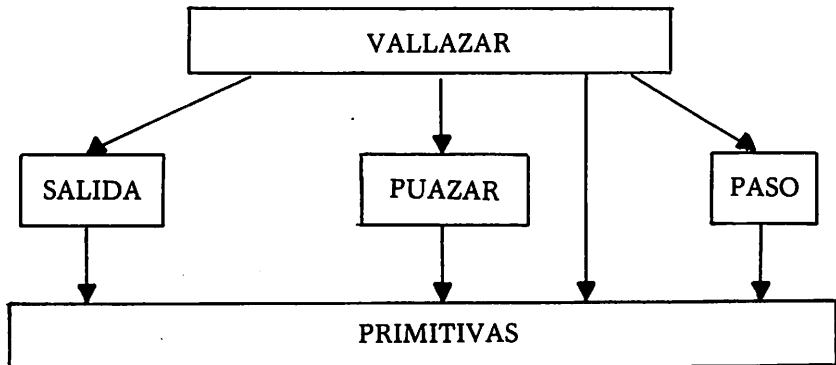
5.2. EJEMPLOS DE JUEGOS.

Los procedimientos que siguen han sido realizados por nuestros alumnos de la Escuela Universitaria de Magisterio de Badajoz.

5.2.1. Juego de Vallas.

Se trata de colocar o recoger emisores en distintos puntos de las vallas. Se pueden dibujar una serie de vallas al azar, por medio del procedimiento VALLAZAR a partir de otras tres (SALIDA, PUAZAR y PASO).

El procedimiento VALLAZAR se implementa en el esquema 5.

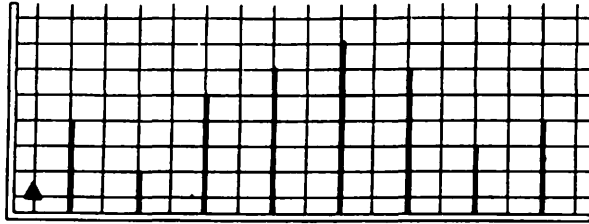


```
PR VALLAZAR
BG KAREL AT 10
CP PONPC 5
DE 90 AD 14 IZ 90
REPITE 8 [PUAZAR PASO]
PONPC 1
SALIDA
FIN
```

```
PR PUAZAR
CP PONPC 5
HAZ "A AZAR 6
AD 25 + (15 * :A)
DE 90
AD 2 IZ 90
AT 25 + (15 * :A)
PONPC 1
FIN
```

```
PR IN :I
HAZ :I SUMA 1 COSA :I
FIN
```

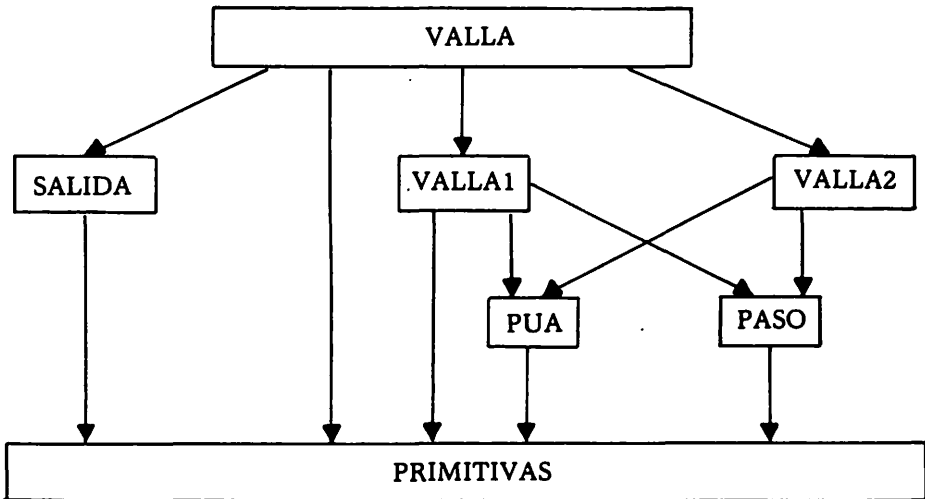
PR PASO
SF
DE 90
AD 28 IZ 90
CF
FIN



También se puede dibujar una valla en concreto, a partir del procedimiento VALLA que consta, a su vez, de los procedimientos SALIDA, VALLA1, VALLA2, PUA y PASO.

Igualmente se pueden confeccionar independientemente dos tramos de vallas utilizando los procedimientos VALLA1 y VALLA2.

El procedimiento VALLA se implementa en el esquema 6.

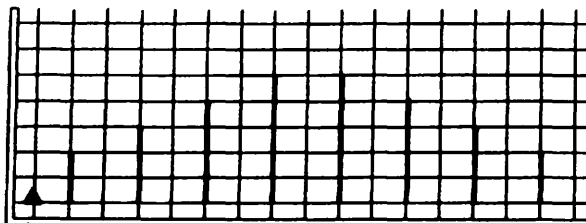


PR PUA :N
AD (15 * :N) + 15
DE 90
AD 2 IZ 90
AT (15 * :N) + 15
FIN

PR VALLA2
CP POMPC 4
HAZ *N 4
REPITE 4 [PUA :N DEC *N PASO]
FIN

PR VALLA1
CP POMPC 4
DE 90 AD 14 IZ 90
HAZ *N 1
REPITE 4 [PUA :N IN *N PASO]
FIN

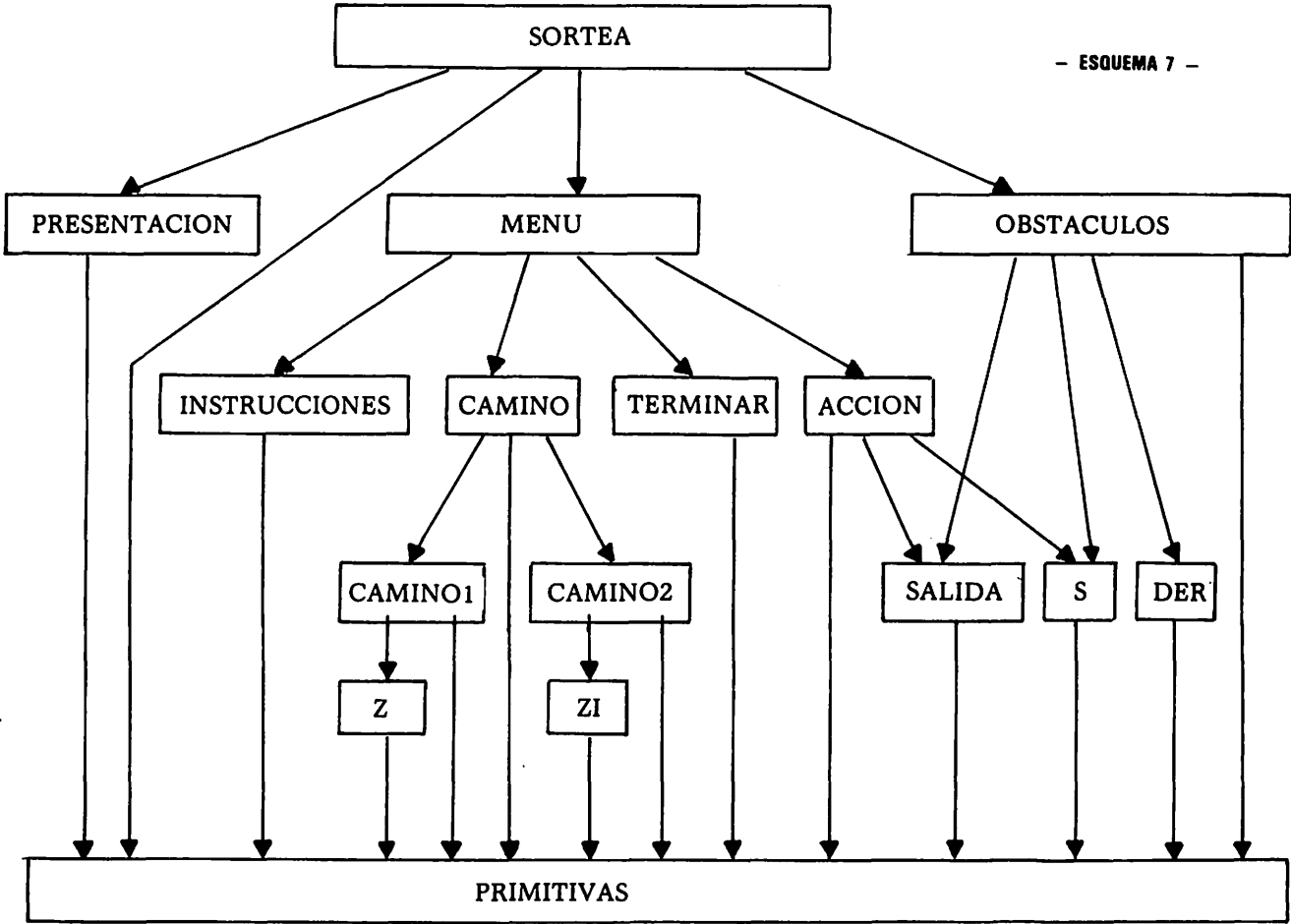
PR VALLA
SALIDA
VALLA1
VALLA2
SALIDA
FIN



PR PASO
SP
DE 90
AD 28 IZ 90
CP
FIN

5.2.2. Juego Sortea.

Para finalizar, incluimos el juego SORTEA que consiste en evadir una serie de obstáculos. El esquema del juego presentado es el siguiente:



PR ZI
REPITE 2 [AD 30 IZ 90]
REPITE 2 [AD 30 DE 90]
FIN

PR Z
REPITE 2 [AD 30 DE 90]
REPITE 2 [AD 30 IZ 90]
FIN

PR SORTEA
PRESENTACION
KAREL
OBSTACULOS
ES [PULSA UNA TECLA PARA CONTINUAR]
HAZ *P LC
MENU
FIN

PR PRESENTACION
MODOTEXTO BORRATEXTO
ES [JUEGO: SORTEA LOS OBSTACULOS]
ES [_____]
REPITE 2 [ES []]
ES [ESTE JUEGO SE DESARROLLA EN EL MUNDO DE] ES [KAREL.]
ES [] ES [APARECEN 8 OBSTACULOS QUE DEBES VADEAR] ES [CON LA TORTUGA.]
ES [] ES [TIENES EN EL MENU UNA POSIBLE FORMA DE] ES [VADEAR
AUTOMATICAMENTE CON SOLO PULSAR] ES [LA TECLA '2'.]
ES [] ES [PUEDES CONSULTAR LAS INSTRUCCIONES EN] ES [EL MENU.]
REPITE 2 [ES []]
ES [AHORA VAS A VER A KAREL EN SU MUNDO] REPITE 3 [ES []]
ES [PULSA UNA TECLA PARA CONTINUAR]
HAZ *TECLA LC
FIN

PR OBSTACULOS
SALIDA
HAZ *E 8
S DER S
REPITE 8 [SOLTAREMISOR S S]
SALIDA
FIN

PR MENU
MODOTEXTO BORRATEXTO
PONCURSOR [1 1]
ES [MENU]
ES [____]
REPITE 3 [ES []]
ES [1. VER INSTRUCCIONES] ES []
ES [2. VADEO AUTOMATICO] ES []
ES [3. VADEO COMANDADO DIRECTAMENTE POR] PONCURSOR [3 11] ES
[EL USUARIO.] ES []
ES [4. TERMINAR]
REPITE 3 [ES []]
ES [PULSA (1 / 2 / 3 / 4)]
HAZ *OPCION LC
SI :OPCION = 1 [INSTRUCCIONES]
SI :OPCION = 2 [MODOMIXTO CAMINO]
SI :OPCION = 3 [ACCION]
SI :OPCION = 4 [TERMINAR STOP]
MENU
FIN

PR CAMINO2
REPITE 4 [Z1]
FIN

PR CAMINO1
REPITE 4 [Z]
FIN

PR CAMINO
PONPC 5 CP
CAMINO1
CAMINO2
PONPC 1 SP
PONCURSOR [1 21]
ES [PULSA UNA TECLA PARA CONTINUAR]
HAZ *C LC
MENU
FIN

PR INSTRUCCIONES

MODOTEXTO BORRATEXTO

ES [INSTRUCCIONES]

ES [_____]

REPITE 3 [ES {}]

ES [PULSA D PARA GIRAR A LA DERECHA] ES {}

ES [PULSA I PARA GIRAR A LA IZQUIERDA] ES {}

ES [PULSA A PARA AVANZAR] ES {}

ES [PULSA F PARA FINALIZAR] ES {}

ES [PULSA S PARA SALIDA]

REPITE 3 [ES {}]

ES [PULSA UNA TECLA PARA CONTINUAR]

HAZ "R LC

FIN

PR ACCION

MODOGRAFICO

HAZ "TECLA LC

SI :TECLA = "D [DE 90]

SI :TECLA = "I [IZ 90]

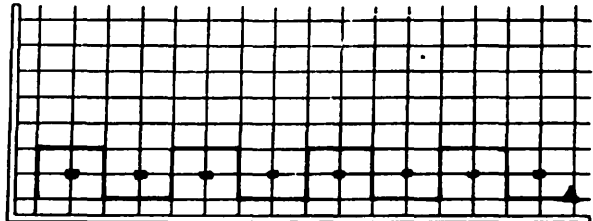
SI :TECLA = "A [S]

SI :TECLA = "F [STOP]

SI :TECLA = "S [SALIDA]

ACCION

FIN



PR TERMINAR

MODOTEXTO BORRATEXTO

PONCURSOR [19 11]

ES (PALABRA CAR 198 CAR 201 CAR 206)

ESPERA 300 BORRATEXTO

FIN

PR S

SALTO

FIN

PR DER

REPITE 3 [IZQ]

FIN

6. CONCLUSIONES.

En general, valoramos nuestro trabajo positivamente debido a que nos ha obligado a reflexionar sobre la Didáctica de la Informática y, en particular, de la Didáctica-LOGO.

Aunque no podemos considerar definitivas estas conclusiones, consideramos que se superaron las dificultades debidas a la existencia de demasiadas primitivas y las propias del teclado.

Creemos que KAREL:

- Puede ser un útil instrumento para "enseñar a pensar" al niño.
- Puede motivarle a plantear y resolver pequeños problemas.
- Puede hacerle reflexionar sobre su propio pensamiento y que comunique al ordenador la manera de resolver el problema de una forma estructurada.

Y, en general, puede constituir un puente hacia LOGO que es el lenguaje de alto nivel con más posibilidades de implantación en nuestro Sistema Educativo.

REFERENCIAS.

- (1) LUENGO GONZALEZ, R. Y OTROS (GRUPO BETA): "Un punto de vista para la introducción de la Informática y la Tecnología del ordenador en las Escuelas Universitarias de Formación del Profesorado de EGB. *Campo Abierto* n. 2 y en su apartado 2 pg. 55 ya se daba cuenta de ella.
- (2) En el artículo citado en ref. 1, pg. 74, hacemos un conjunto de sugerencias para la posible implantación de la Informática ya aludida.
- (3) No en vano sus siglas son el acrónimo de Beginner's All Purpose Symbolic Instruction Code.
- (4) SEYMOUR PAPERT es el inventor del lenguaje LOGO. Profesor de Matemáticas y Educación en el Instituto Tecnológico de Massachusetts. Ha sido discípulo de Piaget y trabajó con Marvin Minsky, Director del Laboratorio de Inteligencia Artificial del M.I.T. Es una autoridad mundial en el tema de Computadoras y Educación.
- (5) S. PAPERT: *Desafío a la mente*. Ed. Galápagos. Buenos Aires 1981. (Pg. 49-53).
- (6) LUENGO GONZALEZ, R. Y OTROS (GRUPO BETA): *Posibilidades de los pequeños ordenadores en la Segunda Etapa de EGB*. IV Plan de Investigación e Innovación Educativa. Badajoz 1983.
- (7) PAPERT en su libro *Desafío a la mente* describe la filosofía pedagógica de LOGO y la posibilidad de las computadoras para cambiar el sistema educativo.
- (8) HAROLD ABELSON en su libro *APPLE LOGO*, pg. IX contraponen esta riqueza de expresión que LOGO posee con otros lenguajes: "la falta de poder de expresión de lenguajes como BASIC dificulta a los principiantes el poder escribir programas simples que, sin embargo, hagan cosas interesantes".
- (9) VICTORIO DE MENDIVELZUA, A. Y OTRA: "El niño preescolar y la modalidad LOGO". *Actas I Congreso Mundial de LOGO (Computadoras en Educación y Cultura)*. Buenos Aires. Septiembre 1983.

- (10) ABELSON, H. *Apple LOGO*. Ed. McGraw-Hill. México 1984. En la pg. 11, segundo párrafo: "Aunque la geometría de la tortuga apareció como parte del lenguaje LOGO, su uso no se limita a LOGO. Otros lenguajes también han incorporado los gráficos de la tortuga, como Smalltalk y Goldberg y UCSD Pascal".
- (11) LUENGO GONZALEZ Y OTROS (GRUPO BETA): "Aprendizaje sintónico de la Geometría con ayuda del ordenador: primeras proyecciones de los ejes de referencia". *Actas IV Jornadas sobre Aprendizaje y Enseñanza de las Matemáticas*. Tenerife. Septiembre de 1984.
- (12) GRUPO BETA (Grupo Extremeño de Didáctica de las Matemáticas) Badajoz. El núcleo principal del Grupo lo constituyen los profesores del departamento de Matemáticas—Física de la E.U. de Magisterio de Badajoz, al que se han unido una psicóloga, un profesor de I.N.B., un biólogo y un profesor de EGB. Se constituyó en Mayo de 1981 y su objetivo principal es el estudio de la Didáctica de las Matemáticas.
- (13) Se han llevado a cabo los trabajos: *Posibilidades de los pequeños ordenadores en la Segunda Etapa de EGB*, IV Plan de Investigación e Innovación educativa. Badajoz 1983. *Informática para profesores de EGB*, V Plan de Investigación e Innovación Educativa. Badajoz 1984, y el proyecto "INAM", actualmente en curso.
- (14) GRUPO BETA. Proyecto "ALCA-84". Se realizó durante cuatro meses del curso 83-84 en una clase de 3.º de EGB en el C.P. de Prácticas Masculino Anejo a la E.U. de Magisterio de Badajoz. Los resultados obtenidos en esta experiencia nos inclinan a pensar que no son arriesgadas estas previsiones.
- (15) Un resumen de este apartado ha sido publicado en Mayo de 1986 en el n.º 5 de la revista oficial de la Asociación Nacional de Logo "LOGO Y EDUCACION". Barcelona 1986.
- (16) KAREL es un autómata simulado por Richard E. Pattis en 1981. Ver su libro *KAREL the robot: A gentle introduction to the art of programming* publicado en Nueva York en el año citado.
- (17) ANGEL MARIN. "El sistema KAREL de iniciación a la programación". INB "Pío Baroja". Irún. *Actas I Jornadas sobre Informática en la Enseñanza*. Barbastro 1984.