KARELEANDO

Jugando a programar

Por: Fernando Guzmán
omijal@gmail.com

El presente documento surge de la inquietud de ofrecer un manual de introducción básico donde el lector pueda de inmediato poder realizar programas en karel.

INSTALACION

Karel no necesita de proceso de instalación, simplemente crea una carpeta con el nombre de KAREL y copias ahí los archivos (el programa lo puedes bajar de www.omijal.org.mx/karelOMI.zip)

INICIAR KAREL

Para ejecutar KAREL es necesario entrar al folder donde grabaste los archivos y dar doble click en el icono de Karel

MENUS

Los menús del Karel son:

Mundos.-

- **Creación.-** Situar a Karel (hacia el norte, sur, este u oeste), colocación de zumbadores (número en específico, 'n' zumbadores o zumbadores infinitos) y creación de paredes se realizar con el 'clic' izquierdo del ratón sobre la mitad de dos calles. Recuerde que un mundo está rodeado por una pared infranqueable.
- **Modificación.** Permite cambiar las posiciones de los zumbadores, quitar zumbadores (para eliminar los zumbadores de una esquina, seleccione colocar 0 zumbadores del menú contextual que aparece al presionar el botón derecho en una esquina del mundo).
- **Guardar y Guardar como.** El mundo es guardado en una carpeta que se le indica con la extensión MDO. Si el mundo ya fue guardado previamente, se guarda con el mismo nombre al seleccionar la primera opción. Guardar como permite archivar el mundo con otro nombre o en otro dispositivo diferente al de origen.
- **Indicar zumbadores en la mochila.** Aquí colocaremos el número de zumbadores que Karel trae inicialmente en la mochila.
- **Viajar a través del mundo.** Los cuatro botones permiten viajar a través del mundo, como este no cabe totalmente en la pantalla, con ellos podemos visualizar otras partes de éste.

Programa.-

- **Nuevo.** Permite inicializar un programa nuevo con las instrucciones esenciales:

```
iniciar-programa
    inicia-ejecucion
        apagate;
    termina-ejecucion
finalizar-programa
```

- **Abrir.** Permite seleccionar un programa para abrirlo y poder trabajar con el.
- **Guardar.** Guarda el programa que se tiene en pantalla con el nombre que se abrió.
- **Guardar como.** Se especifica otro nuevo nombre y/ubicación para el programa.
- **Compilar.** Revisa que el programa que está cargado no tenga errores. Si existen errores nos indicará en que línea el Karel considera que existe un error o no entiende a la(s) instrucción(es).
- **Lenguaje.** Aquí indicaremos con que lenguaje vamos a programar, para que el Karel interprete las instrucciones en este lenguaje. Las dos opciones son: Pascal y Java.

Ejecutar.-

- **Adelante.** Permite continuar un programa detenido.
- **Detener.** Esta opción detendrá el programa en la línea en la cual se está ejecutando, una vez pausado puede ser ejecutado desde este punto o inicializado.
- **Correr.** Inicia la ejecución del programa que está cargado en la memoria y ha sido compilado.
- **Inicializar.** Inicializa el programa en la primer línea y limpia las variables de control a los valores por omisión o iniciales.
- **Zumbadores en la mochila.** Indica la cantidad de zumbadores que Karel llevará en la mochila; esta cantidad deberá ir acorde a las necesidades del programa que se va a ejecutar.
- **Retardo de la ejecución.** Especifica la cantidad de milisegundos que el programa se detendrá o hará al ejecutar cada línea.

Ayuda.- Aquí podremos visualizar la sintaxis de los comandos que soporta Karel con una explicación sencilla, pero útil.

COMANDOS

- **apagate.** Este comando permite apagar a Karel y dejarlo inmóvil, terminando el programa. Es conveniente mencionar que si el programa termina sin apagar a Karel, este primero no tendrá errores.
- **avanza.** Karel avanzará una calle en la dirección hacia donde está orientado, siempre y cuando no exista una pared enfrente.
- **gira-izquierda.** Le indica a Karel, que en la misma posición en que está, gire hacia la izquierda.
- **coge-zumbador.** Karel tomará uno de los zumbadores disponibles en la esquina donde se encuentre; deberá de haber al menos uno.
- **deja-zumbador.** Karel dejará uno de los zumbadores que trae en su mochila; deberá de traer al menos uno.
- **inicio *expresión general [; expresión general]* fin.** Karel ejecutará la o las *expresiones generales* que se encuentren entre los comandos *inicio* y *fin*.
- **si *termino* entonces *expresion1* sino *expresion2*.** Aquí le indicaremos a Karel que si la evaluación de *término* es verdadera, ejecute la *expresión1*; y si esta es falsa, ejecute *expresión2*.
- **mientras *expresión* hacer.** Serie de comandos los cuales van a ser ejecutados *mientras* la expresión sea verdadera o se cumpla. Los comandos se encuentran 'encerrados' con las expresiones *inicio* y *fin*.
- **repetir *expresión_entera* veces *expresión.*** Karel ejecutará a *expresión* tanta veces como *expresion_entera* lo indique.
- **define-nueva-instruccion *identificador* como *expresion*.** Para minimizar el tiempo y eficientar el programa, Karel permite la definición de módulos o procesos los cuales podrán ser llamados como sean identificados, estos constarán de una serie de instrucciones las cuales van a ser ejecutadas cada vez que el proceso sea llamado.
- Los comentarios pueden ser incluidos en el programa y Karel no los tomará en cuenta. Estos deben de ir encerrados en las llaves **{** y **}** ó con **(*** y ***)** para Pascal; y para Java con **/*** y ***/** ó con **//** al inicio.

FUNCIONES BOOLEANAS

- **frente-libre.** Regresa verdadero si el frente hacia el cual está orientado Karel, está libre.
- **frente-bloqueado.** Regresa verdadero si el frente hacia el cual está orientado Karel, está bloqueado (existe una pared).
- **izquierda-libre.** Regresa verdadero si la calle de la izquierda con respecto a la situación de Karel, está libre.
- **izquierda-bloqueada.** Regresa verdadero si la calle de la izquierda con respecto a la situación de Karel, está bloqueada.
- **derecha-libre.** Regresa verdadero si la calle de la derecha con respecto a la situación de Karel, está libre.
- **derecha-bloqueada.** Regresa verdadero si la calle de la derecha con respecto a la situación de Karel, está bloqueada.
- **junto-a-zumbador.** Regresa verdadero si en la esquina de la calle donde está situado Karel, existe al menos un zumbador.
- **no-junto-a-zumbador.** Regresa verdadero si en la esquina de la calle donde está situado Karel, no existe ningún zumbador.
- **algun-zumbador-en-la-mochila.** Permite evaluar si Karel tiene al menos un zumbador en su mochila.
- **ningun-zumbador-en-la-mochila.** Permite evaluar si Karel no tiene ningún zumbador en su mochila.
- **orientado-al-norte.** Regresa verdadero si Karel está orientado hacia el norte.
- **orientado-al-sur.** Regresa verdadero si Karel está orientado hacia el sur.
- **orientado-al-este.** Regresa verdadero si Karel está orientado hacia el este.
- **orientado-al-oeste.** Regresa verdadero si Karel está orientado hacia el oeste.
- **no-orientado-al-norte.** Regresa verdadero si Karel no está orientado hacia el norte.
- **no-orientado-al-sur.** Regresa verdadero si Karel no está orientado hacia el sur.
- **no-orientado-al-este.** Regresa verdadero si Karel no está orientado hacia el este.
- **no-orientado-al-oeste.** Regresa verdadero si Karel no está orientado hacia el oeste.

PRIMER PROBLEMA

PROBLEMA: La Moneda

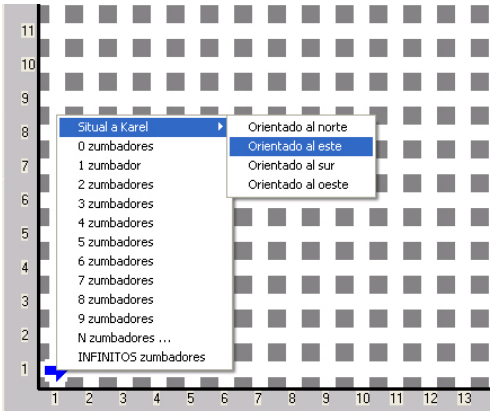
Karel se encuentra en su casa (posición 1,1) viendo hacia el este y se ha dado cuenta que a dos pasos de el se encuentra una moneda, su misión es ir a recogerla y regresar a su casa.

PASOS:

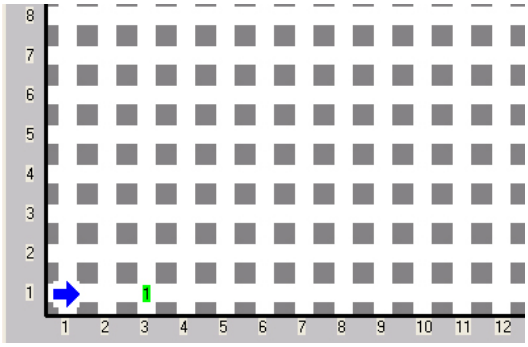
1 - CREAR MUNDO

El primer paso es crear el mundo de prueba donde se ejecutara el programa, para esto ve a la pestaña de MUNDO, da Click en el botón NUEVO.

Ahora colócate en la posición 1,1 , da click al botón secundario de tu mouse y selecciona: Situar Karel – Orientado al ESTE.



Ahora colócate en la posición 1,3 y ahí colocaremos la moneda (1 zumbador), simplemente da click al botón secundario de tu mouse y selecciona 1 zumbador, tu mundo está listo y deberá verse de la siguiente manera:



Por ultimo deberás grabar el mundo, dándole un nombre adecuado, la extensión de los mundos es MDO

2 – PROGRAMA

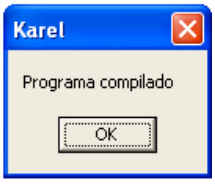
Ahora es momento de hacer el programa que solucione el problema y ayude a Karel a recoger la moneda, ve a la pestaña de PROGRAMA y da click en NUEVO, se creara un programa por default,

```
iniciar-programa
     inicia-ejecucion
          apagate;
     termina-ejecucion
finalizar-programa
```

es momento de iniciar la programación, hazlo escribiendo tus instrucciones después de la línea "inicia-ejecucion", al final, tu código quedaría de la siguiente manera:

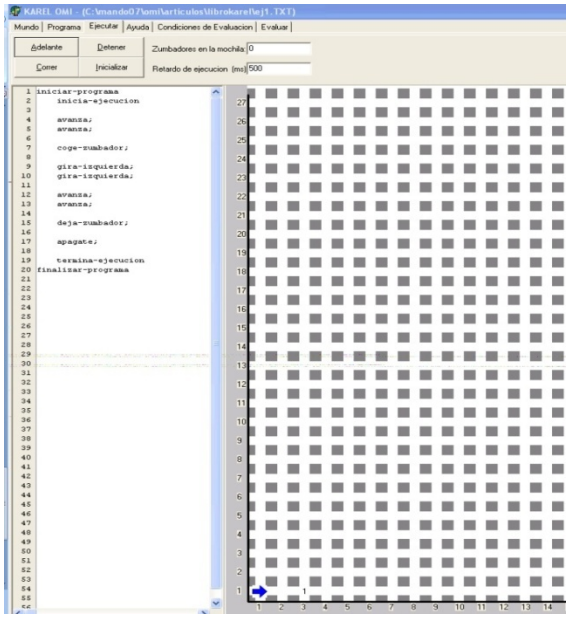
```
iniciar-programa
   inicia-ejecucion
        avanza;
        avanza;
       coge-zumbador;
       gira-izquierda;
       gira-izquierda;
       avanza;
       avanza;
       deja-zumbador;
       apagate;
   termina-ejecucion
finalizar-programa
```

al terminar presiona el botón de "COMPILAR" y de recibir el mensaje de compilación correcta deberás "GUARDAR" tu programa.



3 – EJECUTAR

Tu programa está listo y el mundo creado, ahora es momento de ejecutar y saber si KAREL pudo cumplir con su misión. Da click al botón de "EJECUTAR" y ahí observaras una pantalla dividida, de un lado tu código, del otro tu mundo.



Ahora es momento de ejecutar el programa, presiona "INICIALIZAR" y posteriormente "CORRER" y veras como Karel está siguiendo las instrucciones del programa hasta llegar a una ventana de Terminación normal.

En la ventana de ejecución además existe una caja de texto (Retardo de Ejecución) donde el usuario puede aumentar o disminuir la velocidad de Karel en la ejecución del programa.

Es reconfortante ver correr y funcionar un programa sin error, lo has logrado, pero tenemos que recordarte que este fue un mundo "manual" es decir, si el mundo sufre algún cambio, tu programa ya no servirá, y en Karel, cada problema que se aplica se prueba con 5 o 10 mundos diferentes, por lo tanto la programación "manual" no es nada útil.

Por ello ahora iniciaremos con problemas básicos "multimundos" donde tu programa basado en las consideraciones de cada problema deberá funcionar correctamente en cada uno de los mundos.