Lenguaje de Programación: La instrucción while

José Luis Alonzo Velázquez

Universidad de Guanajuato

Agosto 2010

Principios de su utilidad

A medida que avanzamos en programación, queremos que nuestro robot sea más independiente del mundo que lo rodea, es por eso que nuestra función iterate, está muy limitada, ya que debe conocer de antemano el mundo para determinar la cantidad de veces que necesita realizar un bloque de instrucción con el fin de llevar acabo una tarea o trabajo. Por eso es que Karel cuenta con una util iterador llamado *while*, que funcionare mientras cierta(s) condiciones se cumplan.

Sintaxis de la instrucción while

while(<condición>)
{
<instrucción>
<instrucción>
⋮
<instrucción>
}

IMPORTANTE

Las instrucciones que contienen el **while**, se repiten hasta que la condición sea falsa, sin importar cuantas ocasiones lleva ejecutándose. De este modo, podemos realizar una tarea **mientras** una condición sea verdadera y así ya no tenemos que preocuparnos por saber cuantas veces se debe repetir una instrucción, si no por que la condición se siga o no cumpliendo.

Ejemplo

Si quisieramos que Karel caminara hasta topar con una pared, podriamos usar un **iterate**, pero si no sabemos donde esta la pared, solo que están en algún punto enfrente de Karel, es cuando resulta útil nuestro nuevo iterador.

```
Define camina_a_pared()
        {
        while(frontIsClear)
                move();
        }
```

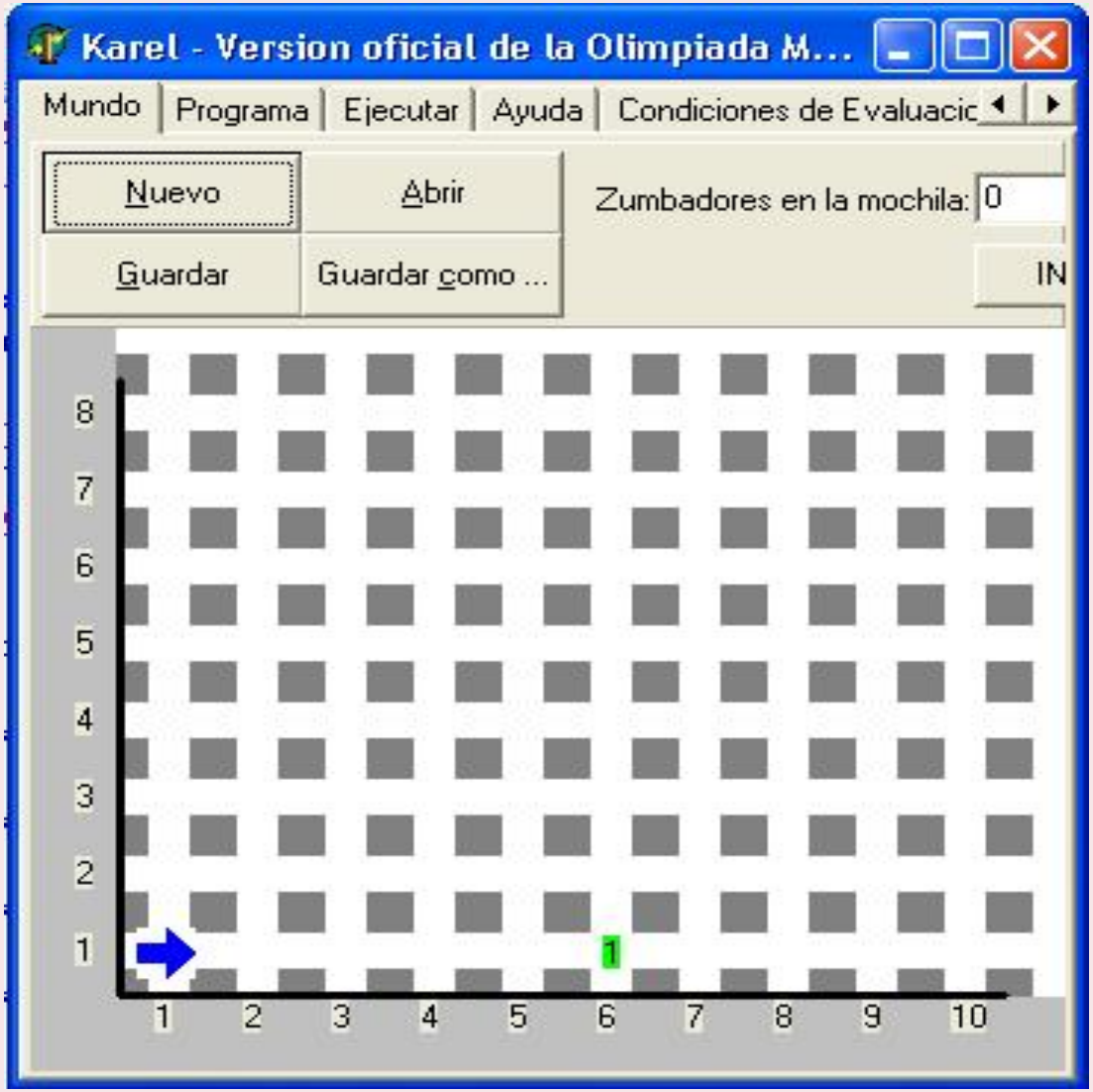
Instrucción while

Al igual que la instrucción iterate, podemos incluir más de una instrucción. Solo necesitamos que las instrucciones esten encerradas entre las llaves.

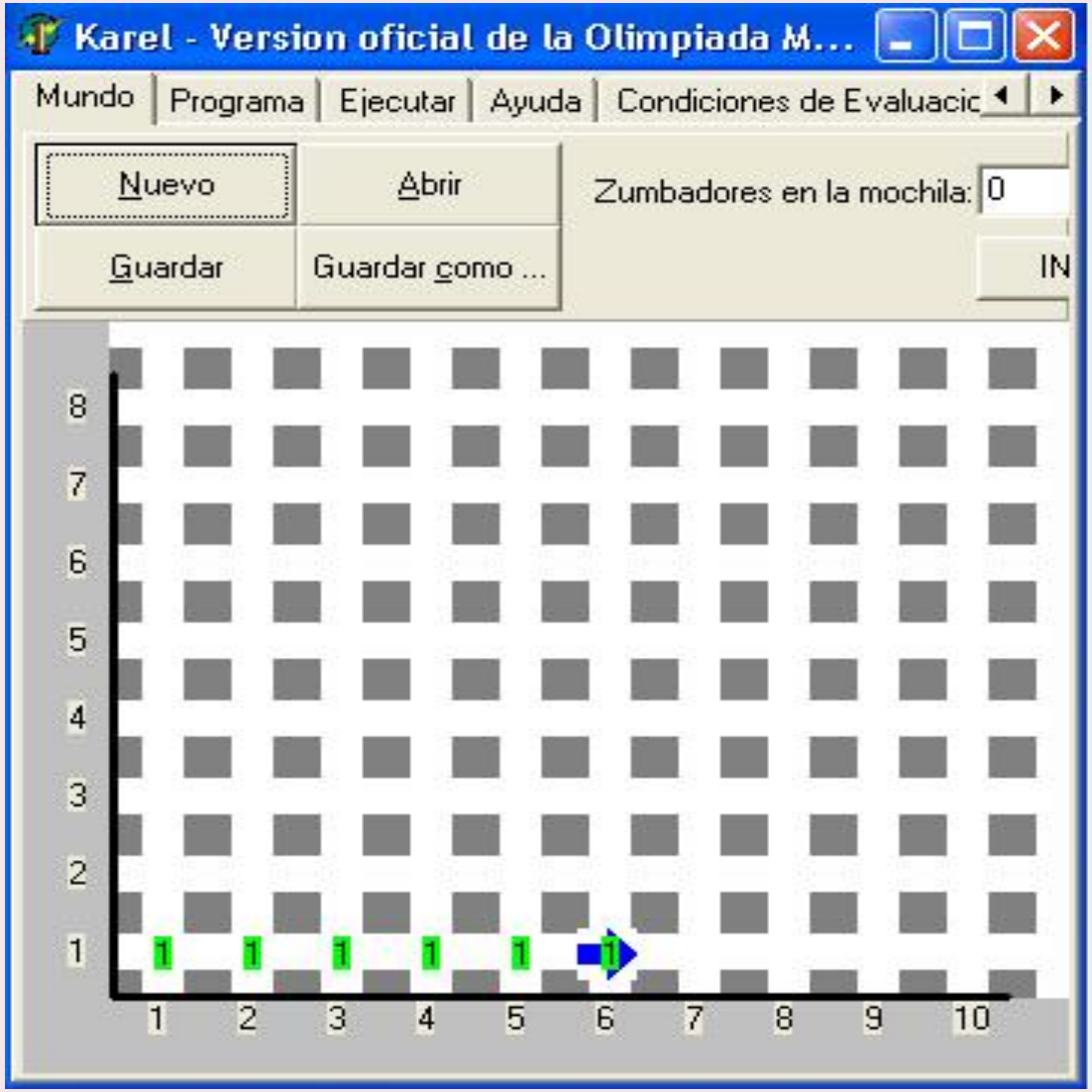
Ejemplo

```
while(notNextToABeeper)
        {
        putbeeper();
        move();
        }
turnoff();
```

Ejemplo posición inicial



Ejemplo posición final

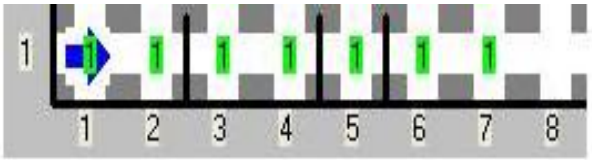


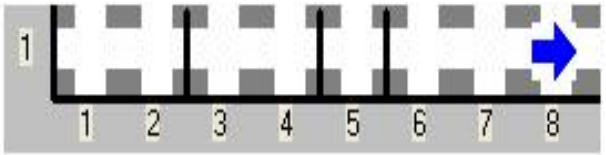
Mirando al norte

```
while(notFacingNorth)
        {
        turnleft();
        }
```

Cosechando la calle

Nuestro querido Karel se encuentra frene a una calle que tiene beepers y obstáculos(de una pared de alto). Los beepers se encuentran uno tras otro, uno en cada esquina. Los obstáculos pueden o no encontrarse entre esquina y esquina. Karel termina cuando encuentra la primera esquina sin beeper.





```
program()
        {
        while( nextToABeeper )
                {
                pickbeeper();
                if(frontIsClear)
                        move();
                else
                        salta_pared();
        }
```

Anidando la instrucción while

Como en los casos de **iterate** y del **if-else**, la instrucción puede anidarse cuantas veces sea necesaria para ejecutar alguna tarea.