

# Ciclo while y arreglos en el lenguaje de MATLAB

**Objetivos.** Programar ciclos while para trabajar con arreglos.

**Requisitos.** Se supone que el estudiante ya tiene experiencia de escribir programas (scripts) en MATLAB o en alguno de sus análogos libres (GNU Octave, Scilab, FreeMat).

En los siguientes ejemplos se trata de GNU Octave, pero casi lo mismo se aplica en MATLAB, Scilab y FreeMat.

Notemos que el lenguaje de MATLAB permite evitar los ciclos en muchas situaciones, pero de todas maneras hay que saber programarlos.

**1. Ejemplo de trabajar con un arreglo por medio del ciclo while.** En cualquier editor crear el siguiente archivo de texto y guardarlo en el formato “texto simple” con el nombre `wa1.m`:

```
a = rand(10, 1);
a(10) = -1;
disp(a);
j = 1;
while a(j) >= 0,
    j = j + 1;
endwhile
disp(j)
```

Abrir GNU Octave, moverse a la carpeta donde está guardado el archivo `e1.m` y ejecutar el siguiente comando:

```
wa1
```

Se recomienda ejecutar el comando `wa1` varias veces y comprender qué está haciendo el programa.

- ¿Cuál es el cuerpo del ciclo? ¿Cuál es la condición del ciclo?
- Complete la frase: mientras se cumple la condición ..., se ejecuta el operador ...
- ¿Cuándo se termina el ciclo?
- Si el ciclo se terminó, ¿qué propiedad tiene el elemento del arreglo con índice `j`?
- ¿Qué propiedad tienen los elementos del arreglo con índices menores a `j`? ¿Por qué?
- Describa qué encuentra el programa.

## 2. Condiciones compuestas.

```
v = [3; -7; 9]
v(2) < 0
v(3) < 0
v(4) < 0
(v(4) < 0) && (4 <= length(v))
(4 <= length(v)) && (v(4) < 0)
```

La última línea de este ejemplo muestra que si el primer argumento de la operación `&&` es falso, entonces el segundo no se calcula.

## 3. Ejemplo.

```
a = rand(4, 1);
disp(a);
j = 1;
while (j <= length(a)) && (a(j) >= 0),
    j = j + 1;
endwhile
disp(j)
```

Guarde el programa en un archivo, ejecútelo varias veces, analícelo y explíquelo.