Ciclos

Laboratorio de Procesamiento de Información Meteorológica y Oceanográfica DCAO - FCEN - UBA

September 5, 2023

Ciclos

Los ciclos o esquemas repetitivos permiten ejecutar repetidamente una parte del código tantas veces como sea indicado. Vamos a trabajar con los siguientes casos:

- **for**: Ejecuta un ciclo una cantidad fija de veces.
- **while**: Ejecuta un ciclo mientras sea verdadera una condición.
- **repeat**: Ejecuta un ciclo indefinidamente (la única forma de detener esta estructura es mediante el comando break).

for

La estructura for ejecuta un ciclo realizando una operación para cada elemento de un conjunto de datos.
La sintáxis del for es la siguiente:

```
for(elemento in objeto) {
operacion_con_elemento
}
```

Con esto, le estamos diciendo a R:
"para **cada elemento** en un **objeto**, realizá la siguiente operación."

for

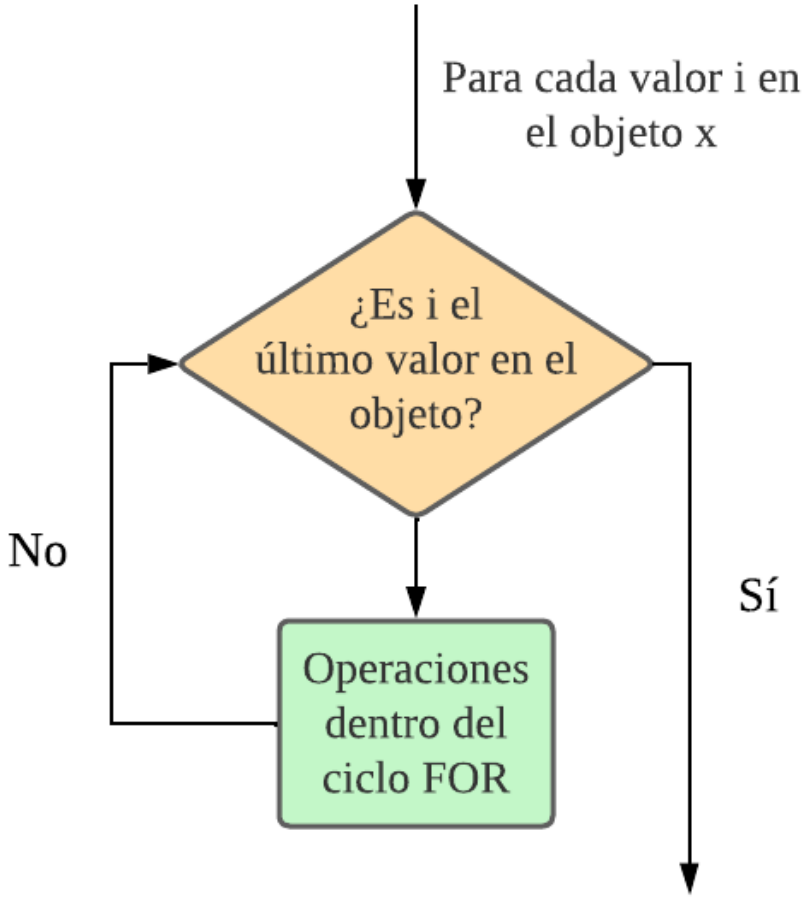

Figure: Diagrama de flujo del ciclo FOR

for

Los nombres que usamos para referir a los elementos es eleccion del programador. Es decir, podemos escribir:

```
objeto <- 1:10
for(elemento in objeto) {
operacion_con_elemento
}
```

secuencia 1,2,3 hasta el 10

recorre todos los elementos del 1 al 10 y realiza la operación

o

```
objeto <- 1:10
for(i in objeto) {
operacion_con_elemento
}
```

i es un contador

recorre todos los i en objetos

Los dos ciclos son equivalentes, sólo cambia el nombre asignado al elemento. Se suelen usar las letras i,j.

for

Vamos a ver algunos ejemplos y predecir que sucedera cuando sean ejecutados en R.
Ejemplo 1:

```
x <- 1:10
# Caso 1:
for (i in x) {
i**2
}
```

for

Ejemplo 2:

```
x <- 1:10
# Caso 2:
for (i in x) {
print(i**2)
}
```

for

```
x <- 1:10
# Caso 2:
for (i in x) {
print(i**2)
}
```

```
[1] 1
4
9
16
25
36
49
64
81
100
```

for

Ejemplo 3:

```
x <- 1:10
# Caso 3:
for (i in x) {
y <- i**2
}
```

for

```
x <- 1:10
# Caso 3:
for (i in x) {
y <- i**2
}
```

for

Ejemplo 4:

```
x <- 1:10
# Caso 4:
for (i in x) {
y <- i**2
assign(paste("y", i, sep = "_"), y)
rm(y)
}
```

assign: asigna a la variable y el nombre de todo lo que esta en paste
A cada y le va asignando ese nombre

for

Que sucede en el Ejemplo 4?
Ejercicio: Generar un ciclo que almacene los valores mismos valores i**2 en un vector. Con i desde 1 hasta 10.

for

Una solucion

```
x <- 1:10
resultados <- c()
# Ejercicio:
for (i in x) {
resultados <- c(resultados, i**2)
}
print(resultados)
```

```
[1] 1 4 9 16 25 36 49 64 81 100
```

while

while es un tipo de ciclo que se repite hasta que una condición sea falsa (FALSE). La operación se realiza hasta que se llega a cumplir un criterio previamente establecido.
La sitaxis del while es la siguiente:

```
while(condicion) {
operaciones
}
```

Con esto le estamos deciendo a R:
MIENTRAS esta condición sea VERDADERA, ejecuta estas operaciones.

while

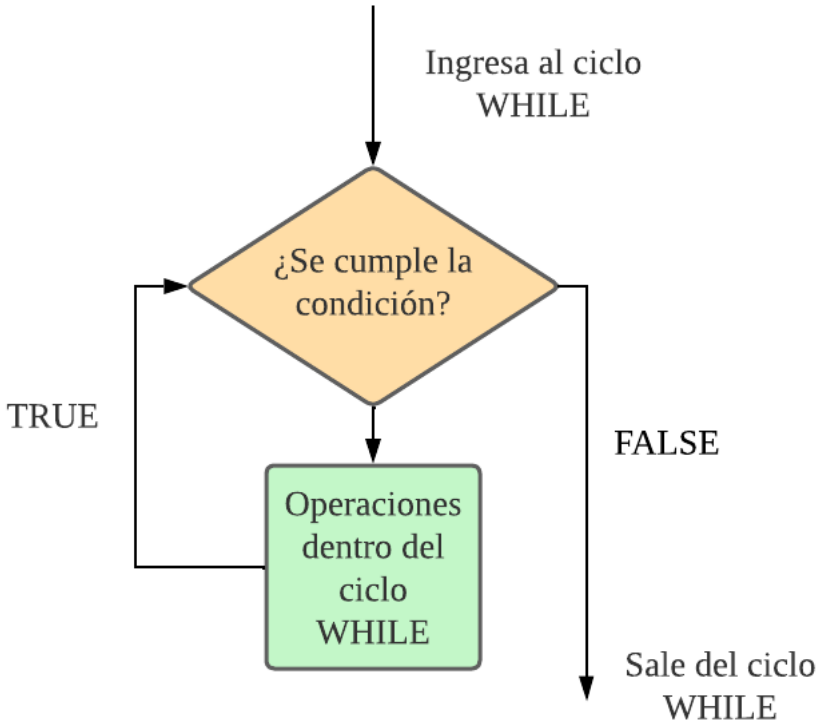

Figure: Diagrama de flujo del ciclo WHILE

while

Ejemplo 1:
Se quiere sumar sumar $+1$ a un valor mientras que éste sea menor que 5.
Al igual que con for, necesitamos la función print() para mostrar los resultados.

```
umbral <- 5
valor <- 0
while (valor > umbral) {
print("Todavia no se ha alcanzado el umbral.")
valor <- valor + 1 redefino valor
}
```

while medio tonto porq nunca se iba a ejecutar

si valor > umbral, el while va a ser eterno y va a imprimir eternamente

si lo cambio por el valor -1, imprime dependiendo del valor de valor

while

Ejemplo 2: ¿Qué pasa en este caso?

```
umbral <- 5
valor <- 0
while (valor > umbral) {
print("Todavia no se ha alcanzado el umbral.")
}
```

while

Ejemplo 2: ¿Qué pasa en este caso?

```
umbral <- 5
valor <- 0
while (valor > umbral) {
print("Todavia no se ha alcanzado el umbral.")
}
```

Se está ejecutando un while con una condición que nunca será falsa, por lo que el ciclo nunca se detendrá.

while

Ejemplo 3:

```
while(1 < 2) {
print("Presiona ESC para detener")
}
```

Ejemplo 4:

me devuelve hasta q llegue al 10

```
i <- 0
while(i < 10) {
i + 1
}
```

break y next

break y **next** next realizan operaciones específicas dentro de los ciclos: break permite interrumpir un ciclo, mientras que next deja avanzar a la siguiente iteración del ciclo, saltándose la actual.

break interumpe el ciclo
next me deja saltear el caso q no me interesa

break

Para interrumpir un ciclo con break, necesitamos que se cumpla una condición. Cuando esto ocurre, el ciclo se detiene, aunque existan elementos a los cuales aún podría aplicarse.

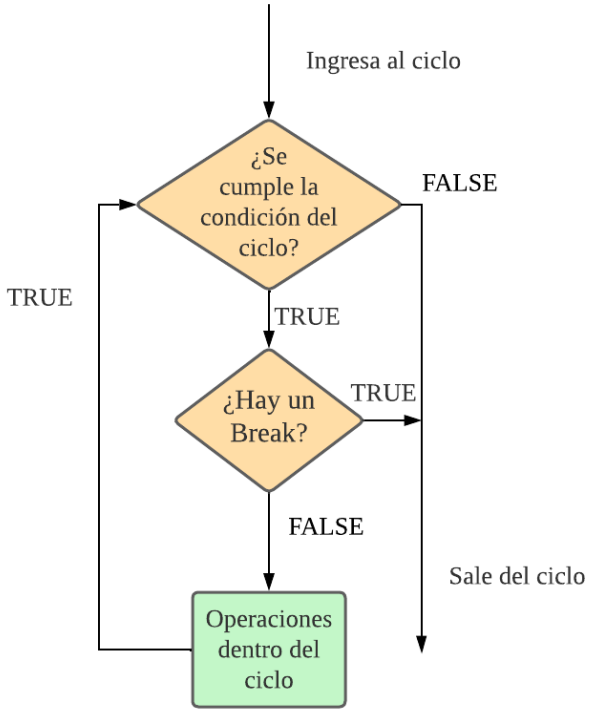

Figure: Diagrama de flujo del ciclo BREAK

break

Dentro de un **for** la sintaxis es la siguiente:

```
for(i in 1:5) {      contador va del 1 al 5
if(i == 2) {         si i=2
break                corta el ciclo
}
print(i)             imprimí los valores de i q llegaste a contar
}
```

```
[1] 1
```

break

Ejemplo:

```
numero <- 19
while(numero > 4) {
if(numero == 14) {
break
}
print(numero)
numero <- numero - 1
}
```

```
[1] 19
 [1] 18
 [1] 17
 [1] 16
 [1] 15
```

next

Se usa **next** para saltear una iteración en un ciclo. Cuando la condición se cumple, esa iteración es omitida.

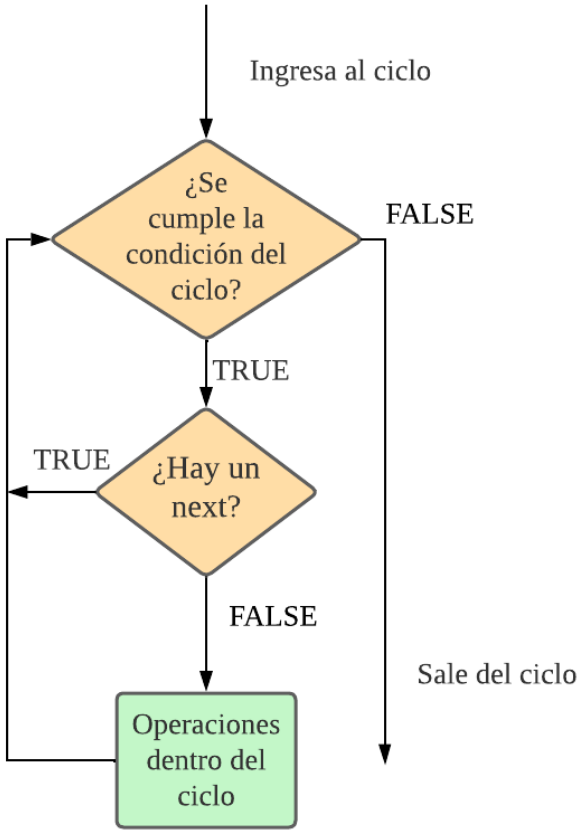

Figure: Diagrama de flujo del ciclo NEXT

next

Se usa **next** para saltear una iteración en un ciclo. Cuando la condición se cumple, esa iteración es omitida.
Dentro de un ciclo for:

```
for(i in 1:5) {
if(i == 2) {
next
}
print(i)
}
```

```
[1] 1
[1] 3
[1] 4
[1] 5
```

repeat

Es un ciclo que se lleva a cabo múltiples veces y se detiene mediante un break. Si no incluimos un break, el ciclo se repetirá indefinidamente. La sintaxis del repeat es la siguiente:

```
repeat {
operaciones
if(condicion){
break
}
}
```

repeat

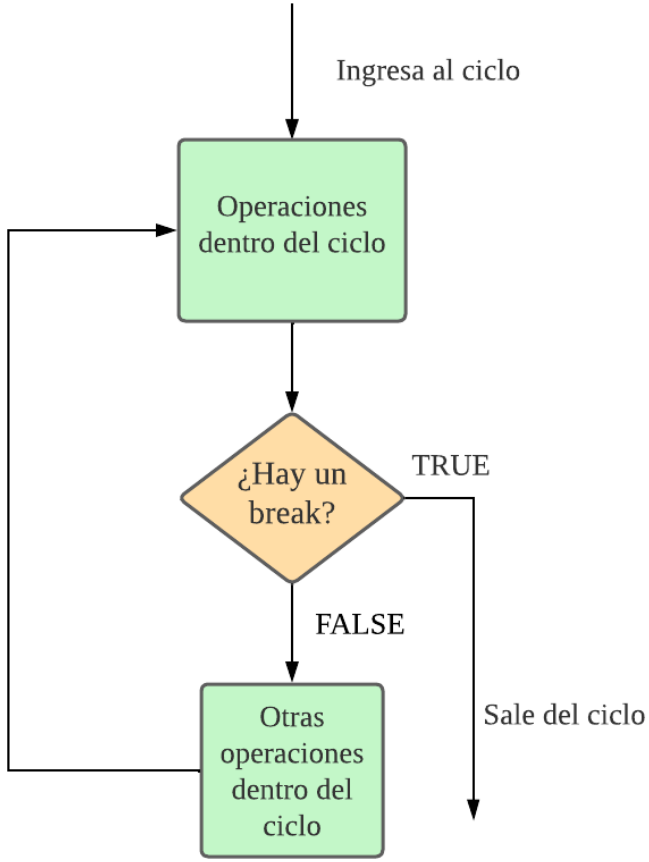

Figure: Diagrama de flujo del ciclo REPEAT

repeat

Ejemplo

```
x <- 1
repeat {
print(x)
x <- x+1
if (x == 6){
break
}
}
```

redefino x porque sino me imprime x=1 infinitamente

ponerle condición porque sino repite infinitamente

```
[1] 1
[1] 2
[1] 3
[1] 4
[1] 5
```

Ejercicio

Ejercicio:
Definir un vector de datos de temperatura en grados Kelvin para un mes de 30 dias donde las temperaturas ascienden 0.5 K por dia desde los 285 K.
Crear un ciclo que convierta a cada uno de los registros en grados Celsius.
Crear un ciclo que, mientras que las temperaturas sean inferiores a los 30°C, imprima el siguiente mensaje por pantalla “El dia X y con una temperatura de XX, todavía no se superó el umbral de los 30°C”.