

Estructuras de control

Cristopher Aguirre

16 de octubre de 2020

1. Ejecución condicional

Cuando se quiere forzar la ejecución de alguna parte del programa según el resultado de la evaluación de una condicional lógica se ejecuta una estructura de control. Hay diferentes estructuras que permiten la ejecución condicional.

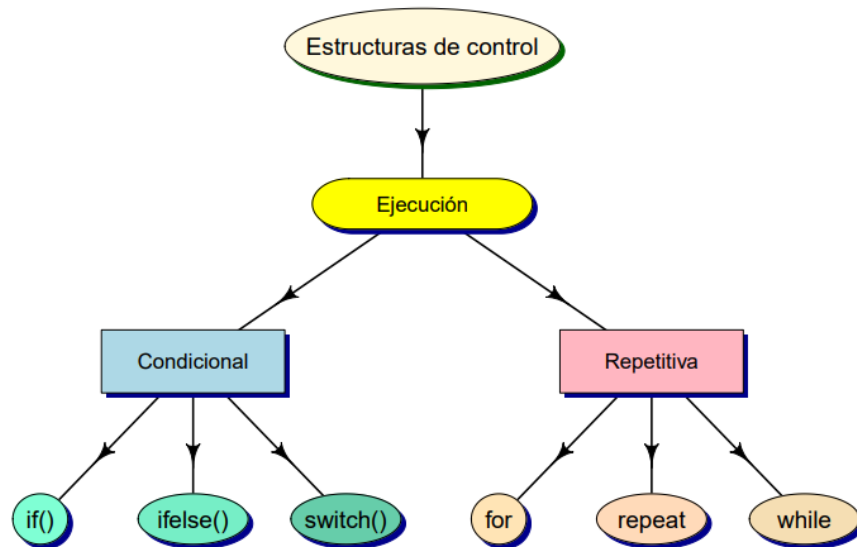


Figure 1: Estructuras de control

1.1. Operadores condicionales if, if else, ifelse

if

Solo evalua una condición lógica y si se cumple se ejecuta un código escrito entre corchetes, si no se cumple salta a la siguiente orden.

```
if (expresion_logica) {  
  expression1 ...  
}
```

Ejemplo

```
a <- 5
b <- 5
if (a == b){
  print("a es igual a b")
}
```

```
## [1] "a es igual a b"
```

if else

Aquí siempre se ejecuta un bloque de código, el bloque uno si se cumple o el bloque dos si no se cumple.

```
if (logical_expression) {
  expression_1
  ...
} else {
  expression_2
  ...
}
```

Ejemplo

```
a <- 7
b <- 5
if (a == b){
  print("a es igual a b")
} else{
  print("a es diferente de b")
}
```

```
## [1] "a es diferente de b"
```

La evaluación lógica no es vectorizada. Dado un vector de números enteros, escriba un programa que diga si son positivos o negativos.

```
x <- c(2, 3, -5, 6, -2, 8)
if (x > 0) {
  print("positivo")
} else {
  print("negativo")
}
```

```
## Warning in if (x > 0) {: la condición tiene longitud > 1 y sólo el primer
## elemento será usado
```

```
## [1] "positivo"
```

ifelse

Es una versión mas corta de if/else en el que se evalúa si cada uno de los elementos de un vector cumple una condicion; si la cumple se adopta la expresión ‘A’ si no la cumple se adopta la expresion ‘B’.

```
ifelse(logical_expression, A, B)
```

Ejemplo

```
x <- c(2, 3, -5, 6, -2, 8)
ifelse(x > 0, "positivo", "negativo")
```

```
## [1] "positivo" "positivo" "negativo" "positivo" "negativo" "positivo"
```

for

Cuando se desea una ejecucion un determinado número de veces, o una secuencia de veces se escribe entre parentesis una variable de ‘vuelta’ y un vector de secuencia. Se ejecutara el código mientras var in seq sea cierto.

```
for(var in seq) expr
```

Ejemplo

```
# Ejemplo 1
for (i in 1:4)
  print(i)
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
```

```
# Ejemplo 2
x <- c(12, 22, 28, 41)
for (i in x)
  print(i+1)
```

```
## [1] 13
## [1] 23
## [1] 29
## [1] 42
```

repeat

Aquí no se utiliza ninguna variable de ‘vuelta’; el código se repite indefinidamente hasta que se apague el ordenador, se pulse Ctrl+C o se encuentre una declaracion **break**.

```
repeat expr
```

Ejemplo

```
i <- 1
repeat {
  if (i > 5) break
  else{
    cat('\n', i, "años de edad")
    i <- i + 1
  }
}
```

```
##
## 1 años de edad
## 2 años de edad
## 3 años de edad
## 4 años de edad
## 5 años de edad
```

while

Otras veces se repite el código mientras una condición es verdadera

```
while(cond) expr
```

Ejemplo

```
i <- 1
while (i<5) {
  i <- i+1
  print(i)
}
```

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