

LABORATORIO DE PROCESAMIENTO DE INFORMACION METEOROLÓGICA

Lenguaje de Programación R

INTRODUCCIÓN A LA LÓGICA DE LA PROGRAMACIÓN

- Expresiones lógicas
- Operadores lógicos
- Operadores relacionales
- Diagramas de Flujo
- Estructuras de control repetitivas (Funciones iterativas FOR, WHILE, REPEAT)
- Estructura condicional simple (Función IF - ELSE - IFELSE, SWITCH)
- Interrupciones de ciclos (BREAK, NEXT, RETURN)

ESTRUCTURA CONDICIONAL

Funciones

IF - ELSE

IFELSE

SWITCH

DIAGRAMA FLUJO IF

si se cumple, ejecuto serie de comandos
sino sigue corriendo el programa

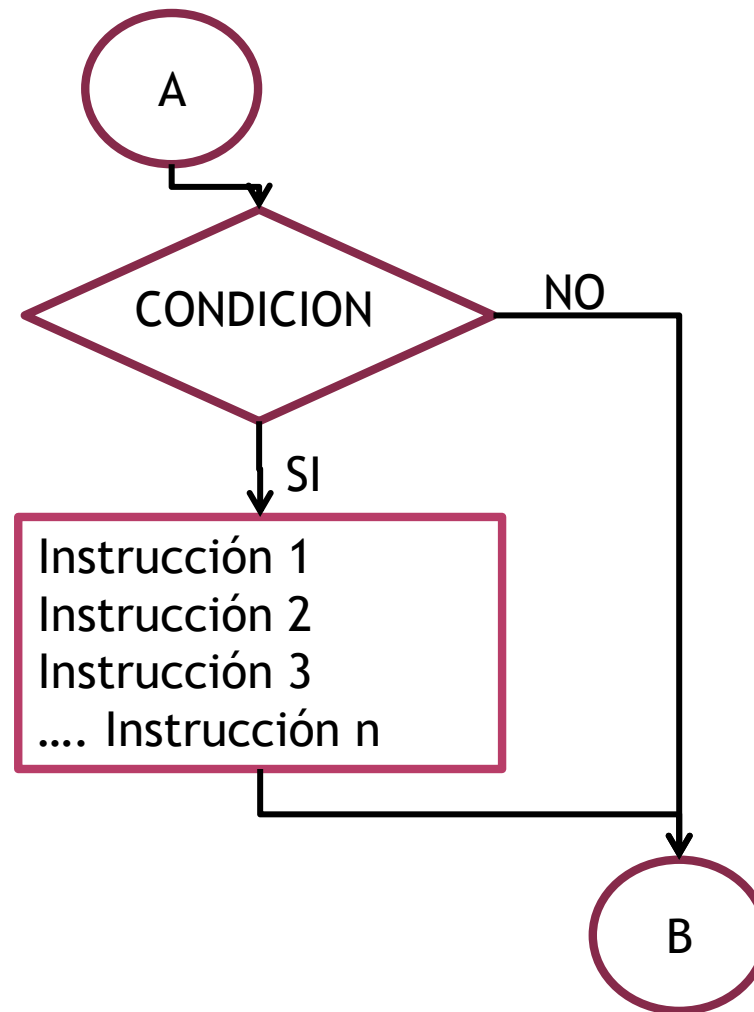
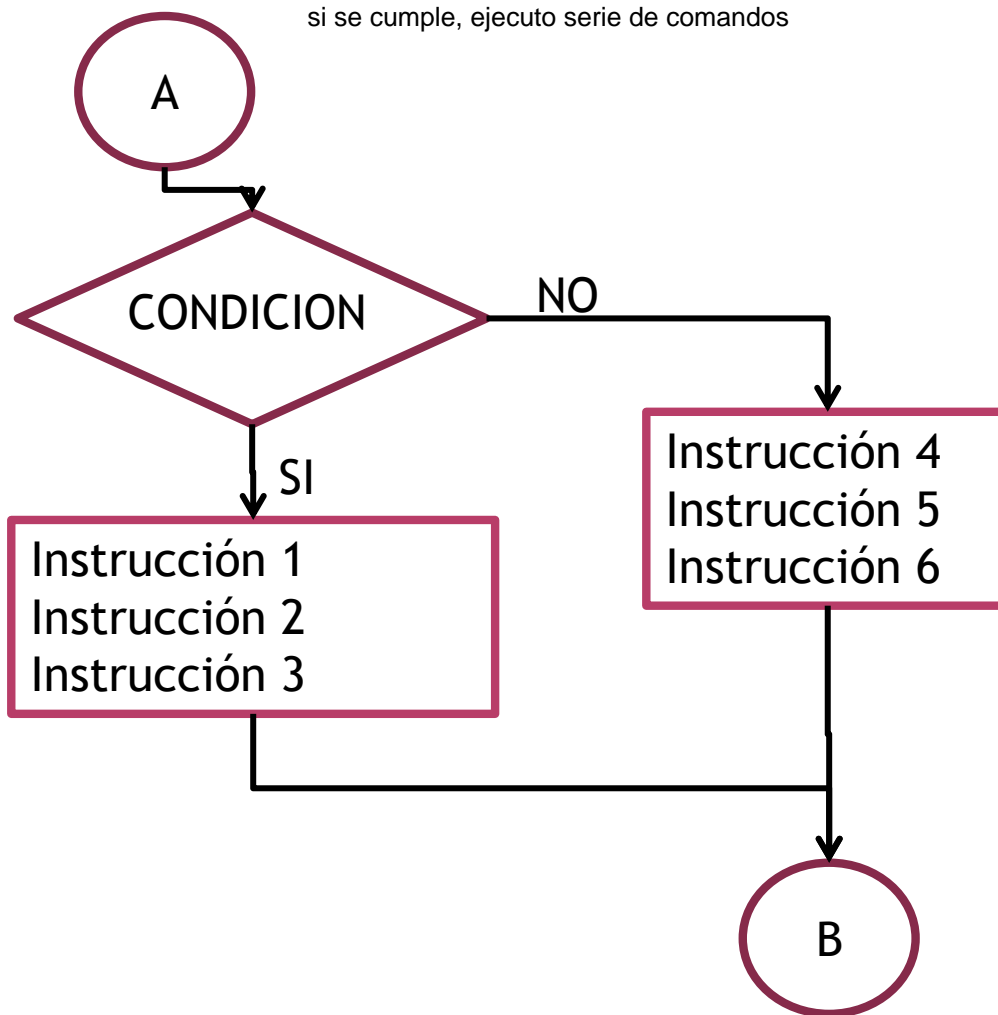


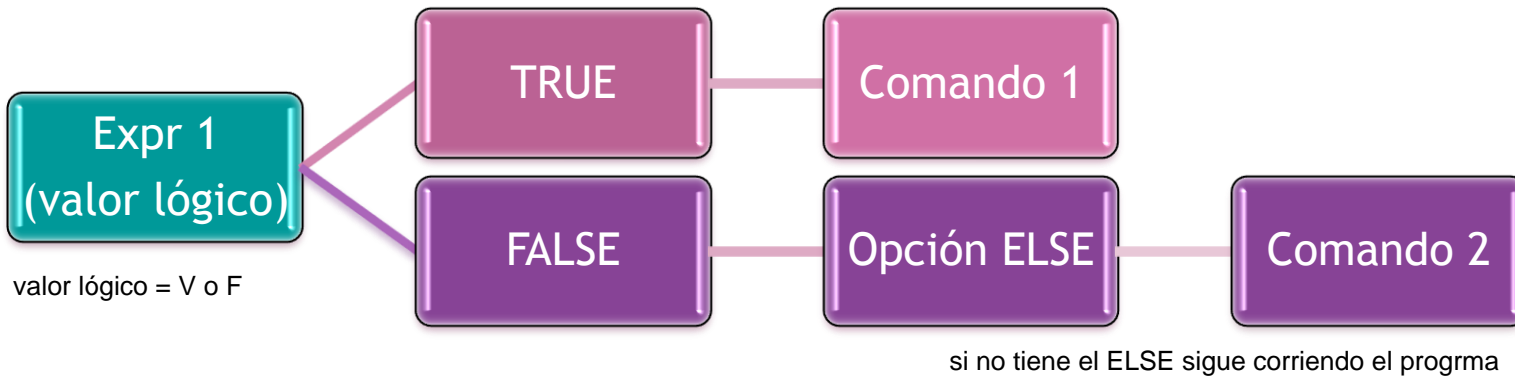
DIAGRAMA FLUJO IF - ELSE



FUNCIÓN IF - ELSE

exp 1 es la condición que tiene que cumplirse

Existe una construcción condicional de la forma
if (expr 1) comando 1 else comando 2

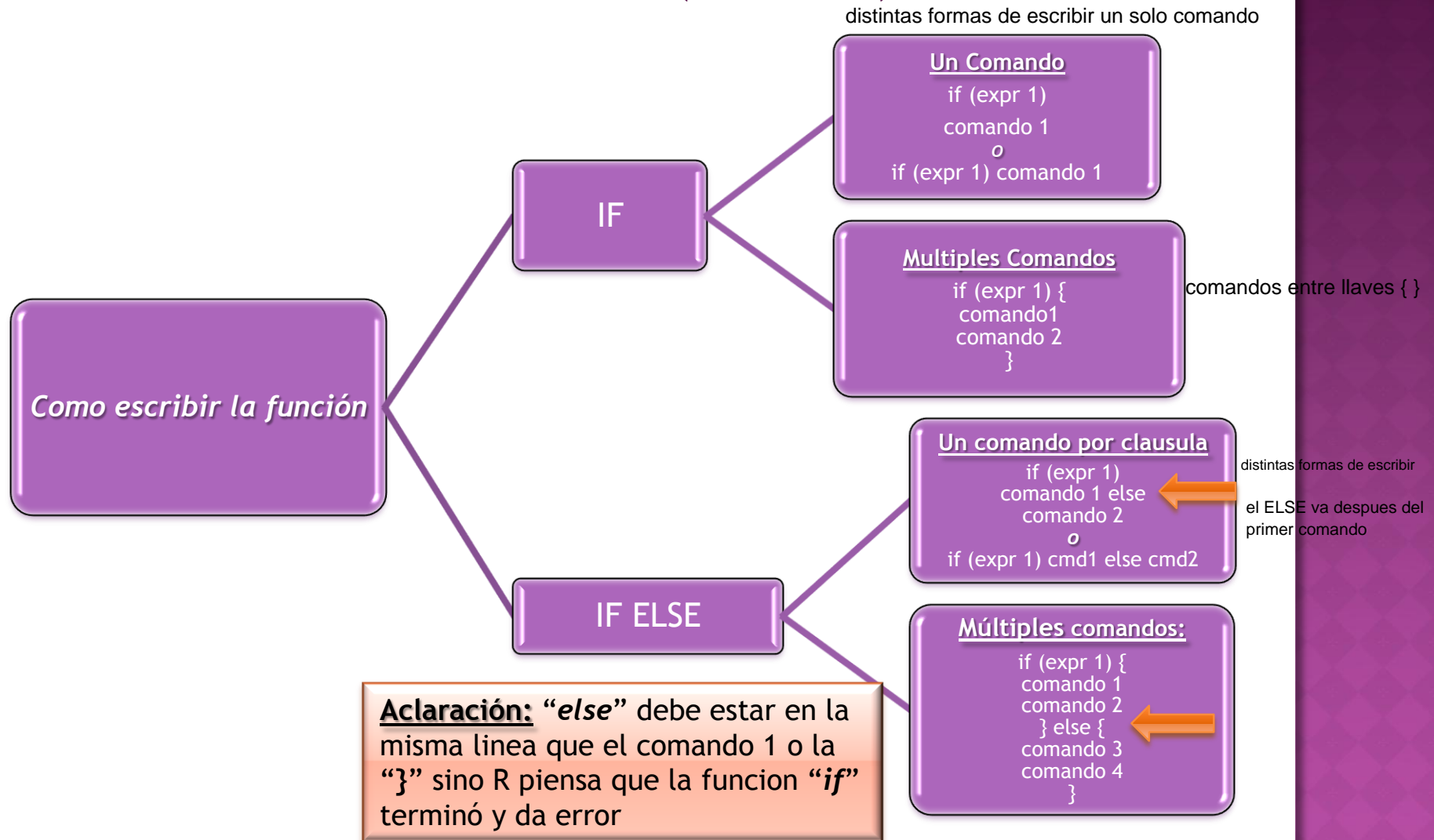


```
rm(list=ls())      # borro las variables existentes
a=pi               # defino nuevas variables
b=18
if a>b             #comienzo a ejecutar la función IF
Error: unexpected symbol in "if a"  #ERROR!!!! La expr 1, a>b, debe estar entre ()
if (a>b)
+                 #aparece el simbolo más para que continúe escribiendo la función
+ c=a+b           para poner la condición
                  #Termino la función IF, si quería continuar no me deja. Resultado?

C                 me da una función
function (... , recursive = FALSE) .Primitive("c") no se cumple la condición entonces no ejecuta la sentencia
```

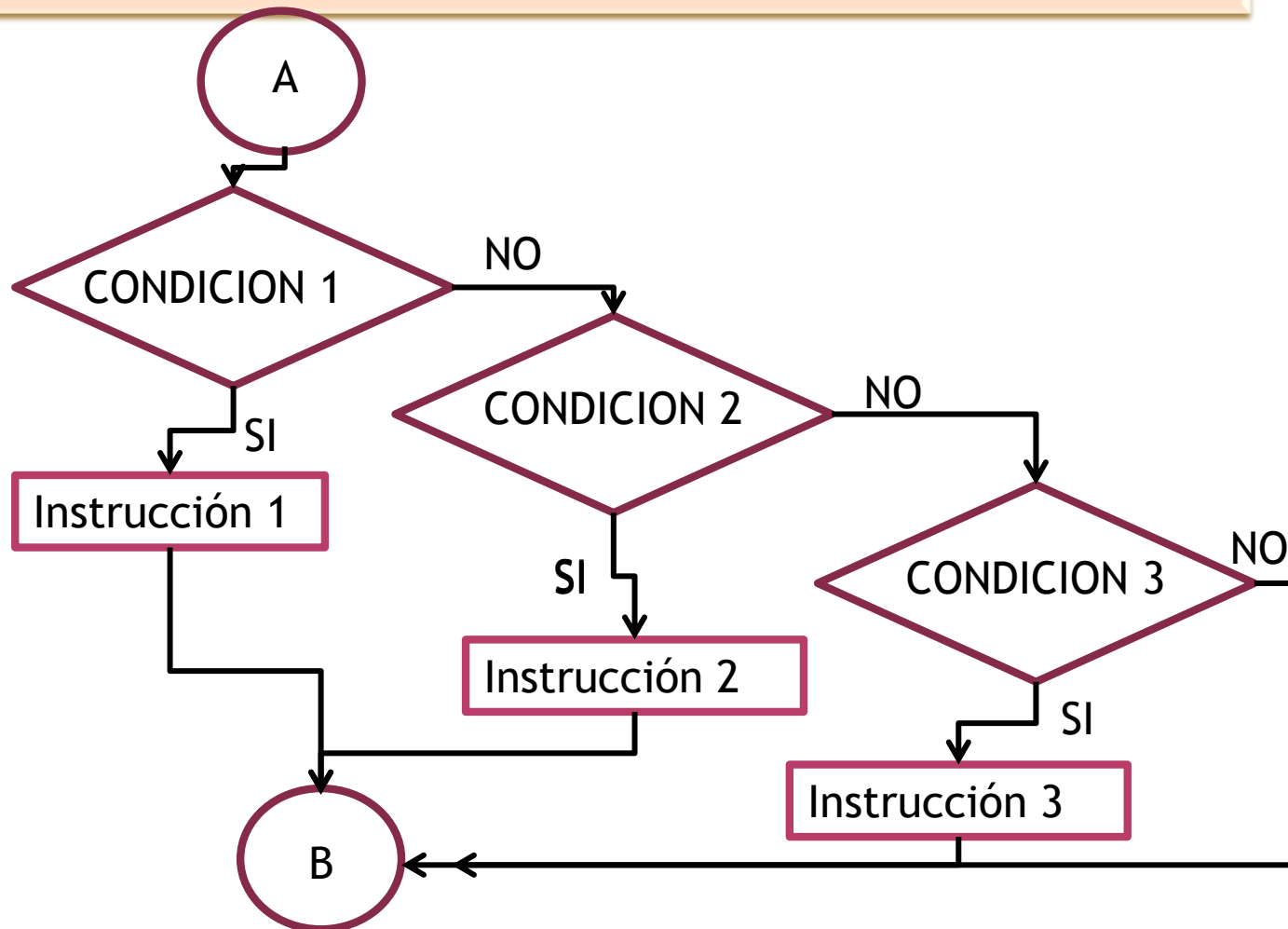
{ } encierran los comandos que me hablan de una condición

FUNCIÓN *IF - ELSE* (CONT)



FUNCIÓN *IF - ELSE* (CONT)

Se pueden escribir funciones *IF/ELSE* dentro de una función *IF/ELSE*



FUNCIÓN *IF - ELSE* (CONT)

para poner if else dentro de otro if else

```
if (expr 1) {  
  comando 1  
} else if (expr 2) {  
  comando 2  
  comando 3  
} else if (expr 3) {  
  comando 4  
} else {  
  comando 5  
}
```

FUNCIÓN *IF - ELSE* (CONT)

Las expresiones pueden ser LÓGICAS

```
x=a>0
```

```
x
```

```
[1] TRUE
```

```
if (x) resultado="entre aca"
```

```
resultado
```

```
[1] "entre aca"
```

a era *pi*, luego al ser mayor que cero *x* es verdadero.
Expresión lógica

```
x=a<0
```

```
if (x) resultado="entre aca"
```

no debería haber dado eso. Hay que tener cuidado con los valores de las variables

```
resultado
```

```
[1] "entre aca"
```

Cambio el valor de *x*

Es cierto esto ??????

```
x
```

```
[1] FALSE
```

```
rm (resultado)
```

```
if (x) resultado="entre aca"
```

```
resultado
```

Elimino el resultado anterior y
Vuelvo a correr la expresión

Error: object 'resultado' not found


FUNCIÓN *IFELSE*


Evalúa la condición. Si es verdadera

Existe una versión vectorizada de la construcción if/else, que es la *función ifelse*

ifelse (condición, a, b)

Evalúa la “condición”, si es VERDADERA entonces devuelve el parámetro “a” de lo contrario devuelve el parámetro “b”


ifelse(T, "el parámetro del test es verdadero", "el parámetro del test es falso")
[1] "el parámetro del test es verdadero"


ifelse(F, "el parámetro del test es verdadero", "el parámetro del test es falso")
[1] "el parámetro del test es falso"

x<-c(T,F,T)
x
[1] TRUE FALSE TRUE

ifelse(x, "el parámetro del test es verdadero", "el parámetro del test es falso")
[1] "el parámetro del test es verdadero" "el parámetro del test es falso"
[3] "el parámetro del test es verdadero"

un resultado por cada elemento del vector

FUNCIÓN *IF - ELSE - IFELSE* (CONT)

Operadores Lógicos

&	• Intersección (and/y)
	• Unión (or/o)
!	• Negación

En una comparación de dos vectores lo hace miembro a miembro. **Resultado:** Un vector de la dimensión del mayor vector

&&	• Intersección (and/y)
	• Unión (or/o)
!	• Negación

En una comparación de dos vectores considera, en general, solo el primer miembro. **Resultado:** en general un escalar

solo compara el 1 elemento

```
a<-c(2,5,10,8)
b<-c(1,15,0,3)
if(a<5 & b>0) rta="cumple la condición"
```

Warning message:

In if (a < 5 & b > 0) rta="cumple la condición" :

the condition has length > 1 and only the first element will be used

```
(a<5 & b>0)
```

```
[1] TRUE FALSE FALSE FALSE
```

```
(a<5 && b>0)
```

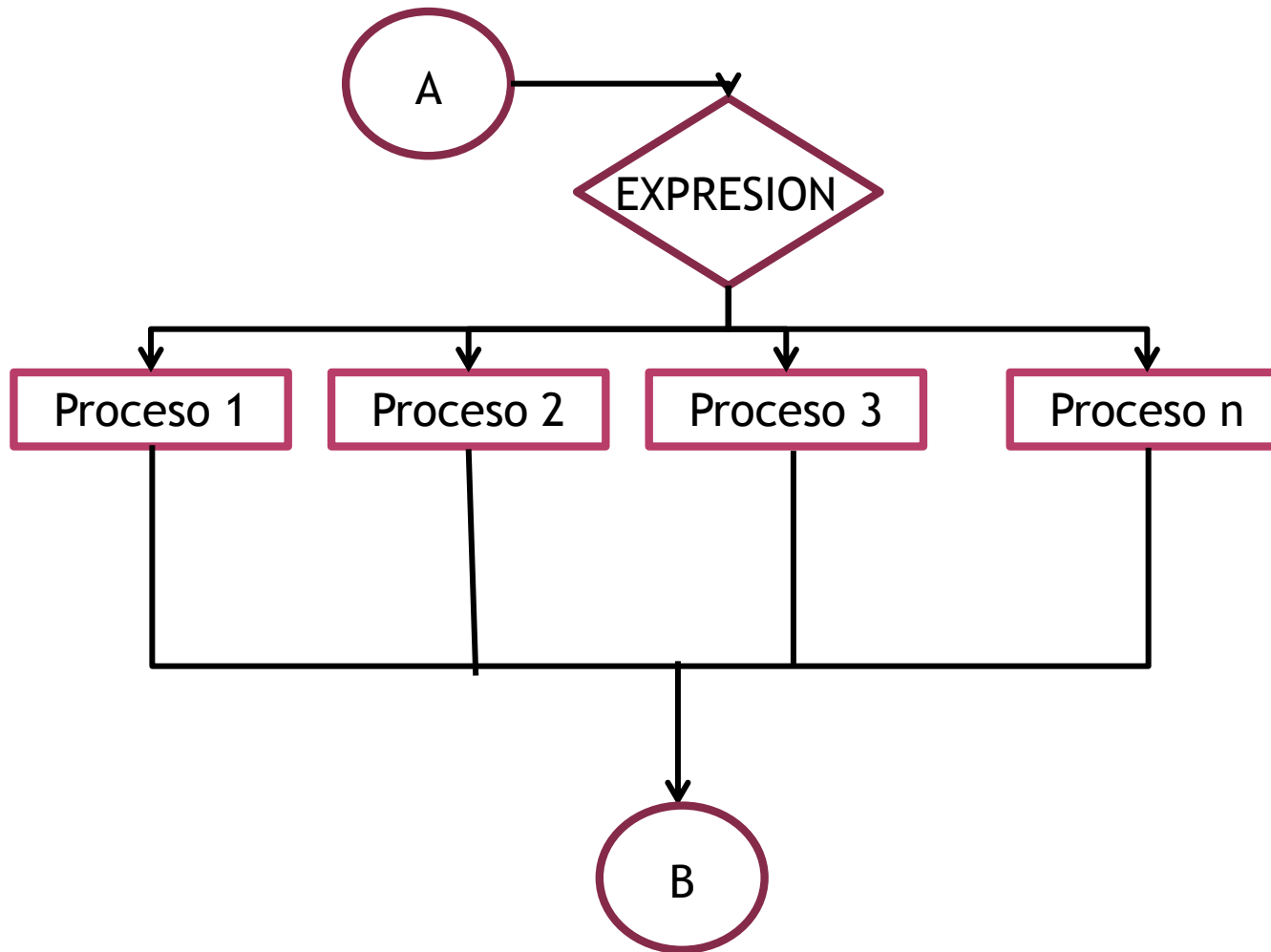
```
[1] TRUE
```

```
if(a<5 && b>0) rta="cumple la condición"
```

Se usa en la función **IF**

NO da mensaje de error y generó la variable **rta**

DIAGRAMA FLUJO - SWITCH



FUNCIÓN SWITCH

switch(EXPR, ...)

Si el valor de **EXPR** no es un caracter se lo convierte en entero.

Si **EXPR** es una variable caracter luego la variable debe coincidir (exactamente) al nombre de los elementos en

Si hay coincidencia luego el elemento es evaluado

En el caso que sea faltante toma el valor del siguiente elemento no ausente

Ejemplo cc y cd no tienen valores asignados

switch("cc", a = 1, cc =, cd =, d = 2)

toma el valor 2.

Si no hay coincidencia, y si hay un elemento no nombrado en ... devuelve ese valor.

```
x<- c(-2.048,1.055,0.832,-4.177,8.298,-0.201,4.630,-3.88,0.938,-1.071)
```

```
type="mean"
```

```
switch(type,mean = mean(x),median = median(x),standdev = sd(x))
```

```
[1] 0.4376
```

```
type="median"
```

```
switch(type,mean = mean(x),median = median(x),standdev = sd(x))
```

```
[1] 0.3155
```

```
type="standev"
```

```
switch(type,mean = mean(x),median = median(x),standdev = sd(x))
```

standev y standdev no escritos igual

entonces no puede ejecutar

type variable caracter

mean -> media

FUNCIÓN SWITCH (CONT)

switch(EXPR, ...)

for ejecuta varias veces la misma sentencia
for pasa por todos los valores de ch y va ejecutando la función

```
ccc <- c("b","QQ","a","A","bb")
```

```
for(ch in ccc)
```

```
  cat(ch,":", switch(EXPR = ch, a = 1, b = 2:3), "\n")
```

```
b : 2 3
```

```
QQ :
```

```
a : 1
```

```
A :
```

```
bb :
```

```
  for(ch in ccc)
```

```
    cat(ch,":", switch(EXPR = ch, a =, A = 1, b = 2:3, "Otherwise: last"), "\n")
```

```
b : 2 3
```

```
QQ : Otherwise: last
```

```
a : 1
```

```
A : 1
```

```
bb : Otherwise: last
```

ch variable

cat concatena elementos

"\n" por cada vez que hace ese calculo lo hace en
el renglón de abajo

NOS EJERCITAMOS...

Para cada uno de los siguientes ejercicios realizar:

- a) El diagrama de flujo
- b) La escritura del código en R

1. Testear si π^e es mayor o igual que e^π (uso de IF)
(ver en Ejemplo 3.R la escritura con formato)

2. Dada la variable x , cuyo valor es *cero*, verificar si la variable es mayor, menor o igual a cero (uso de IF ELSE)

3. Modificar el programa de 2 para que x tome un valor asignado por el usuario (uso de IF ELSE scan)

4. Modificar el programa 3 para que además indique si el valor es par o impar (uso de IF ELSE scan)