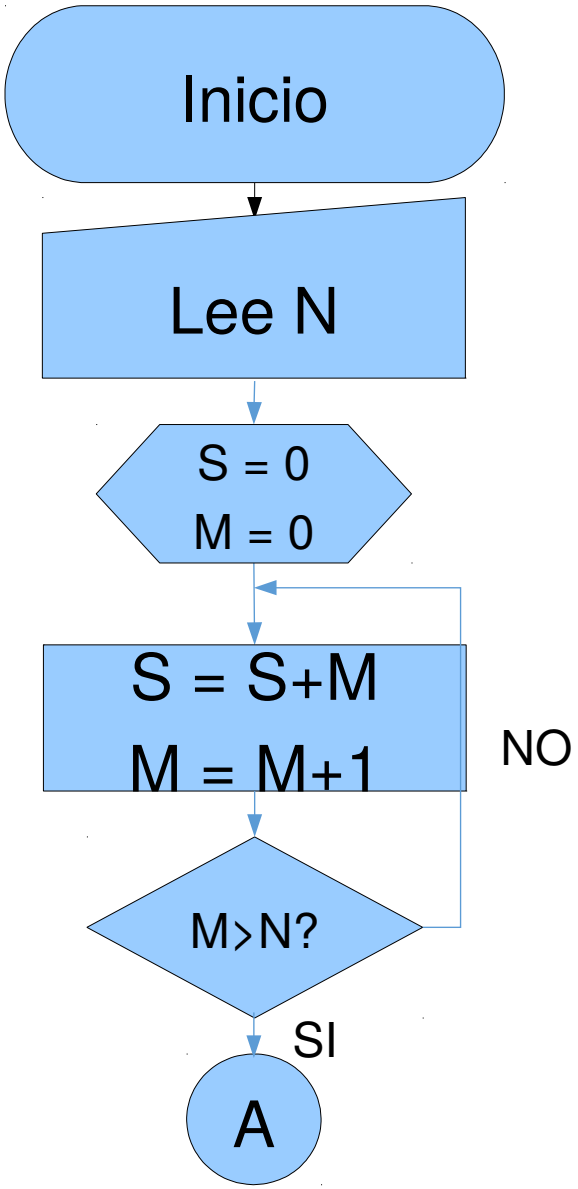
M1964 – Panorama

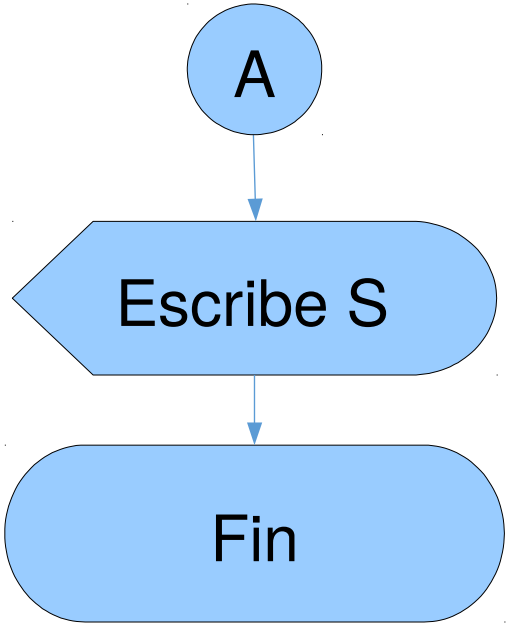
Introducción a R
Sentencias de Control

Sentencias de Control



El conjunto de operaciones nos permitiría únicamente operar de forma secuencial, es decir, **todas las sentencias o instrucciones que introducimos en el código se ejecutan una por una y de arriba abajo**.

Sin embargo, hemos visto que hay otro tipo de estructuras, como los ciclos y las bifurcaciones.



Operadores Básicos: Relacionales

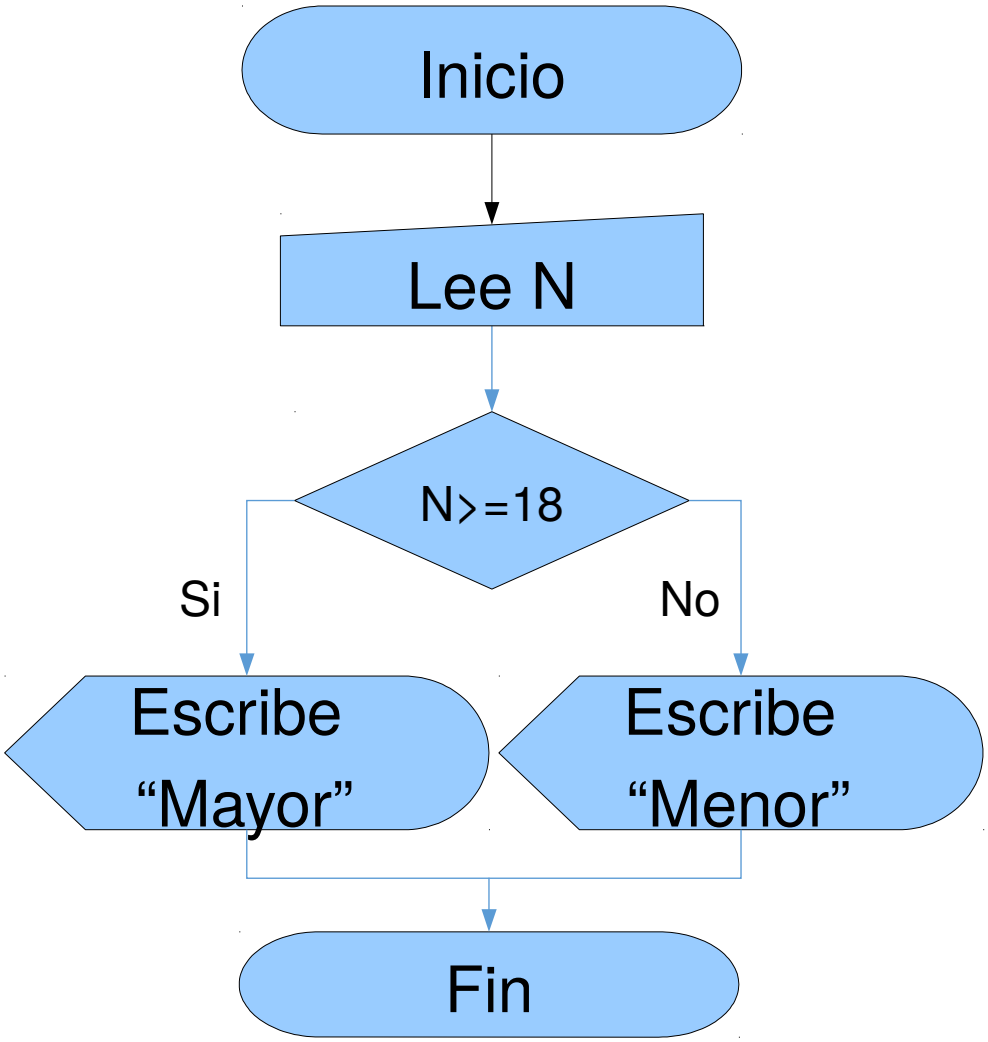
Operación	Símbolo	Ejemplo	Resultado
Menor que	<	5 < 3	
Mayor que	>	5 > 3	
Menor o igual que	<=	5 <= 3	
Mayor o igual que	>=	5 >= 3	
Distinto que	!=	5 != 3	
Igual que	==	5 == 3	

Operadores Básicos: Relacionales

Operación	Símbolo	Ejemplo	Resultado
Menor que	<	5 < 3	*false*
Mayor que	>	5 > 3	*true*
Menor o igual que	<=	5 <= 3	*false*
Mayor o igual que	>=	5 >= 3	*true*
Distinto que	!=	5 != 3	*true*
Igual que	==	5 == 3	*false*

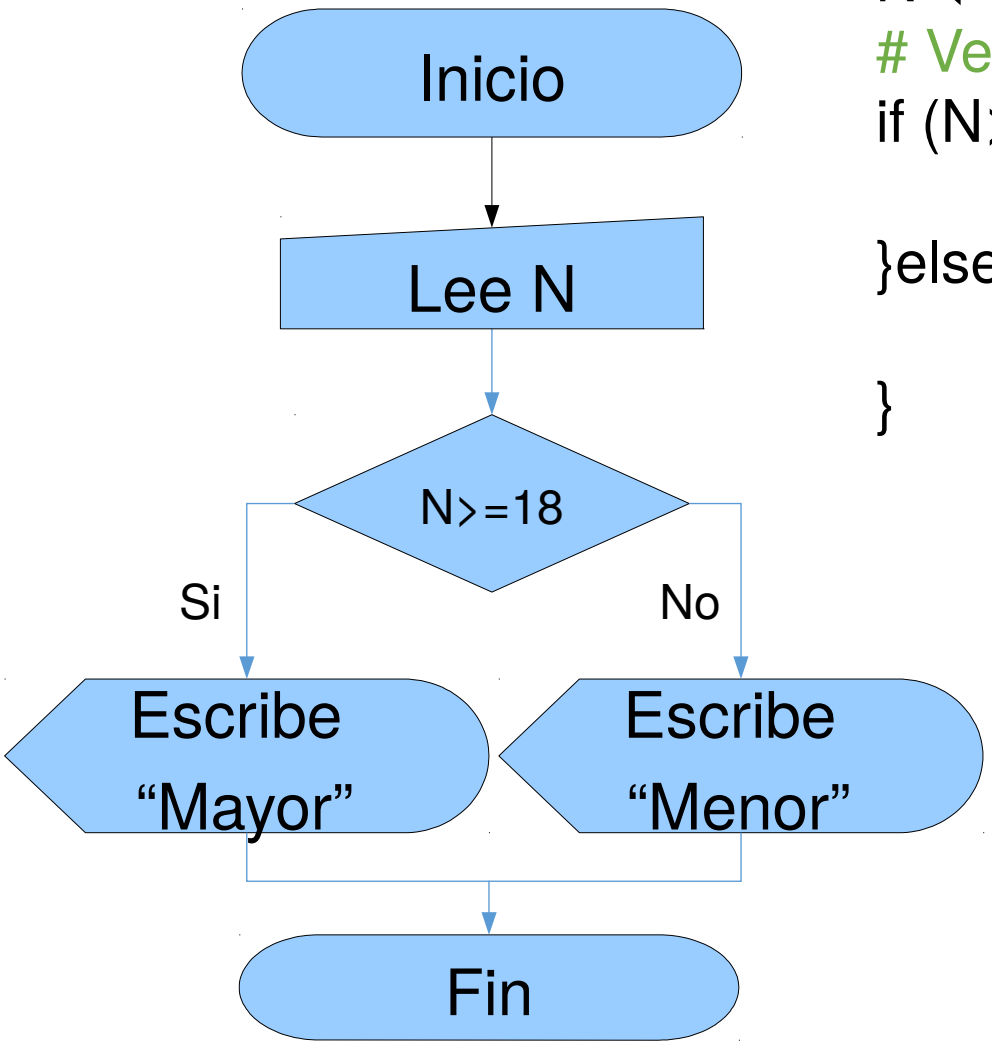
Mayor de Edad

Hacer un programa que solicite la edad y diga si es mayor o menor de edad:



Mayor de Edad

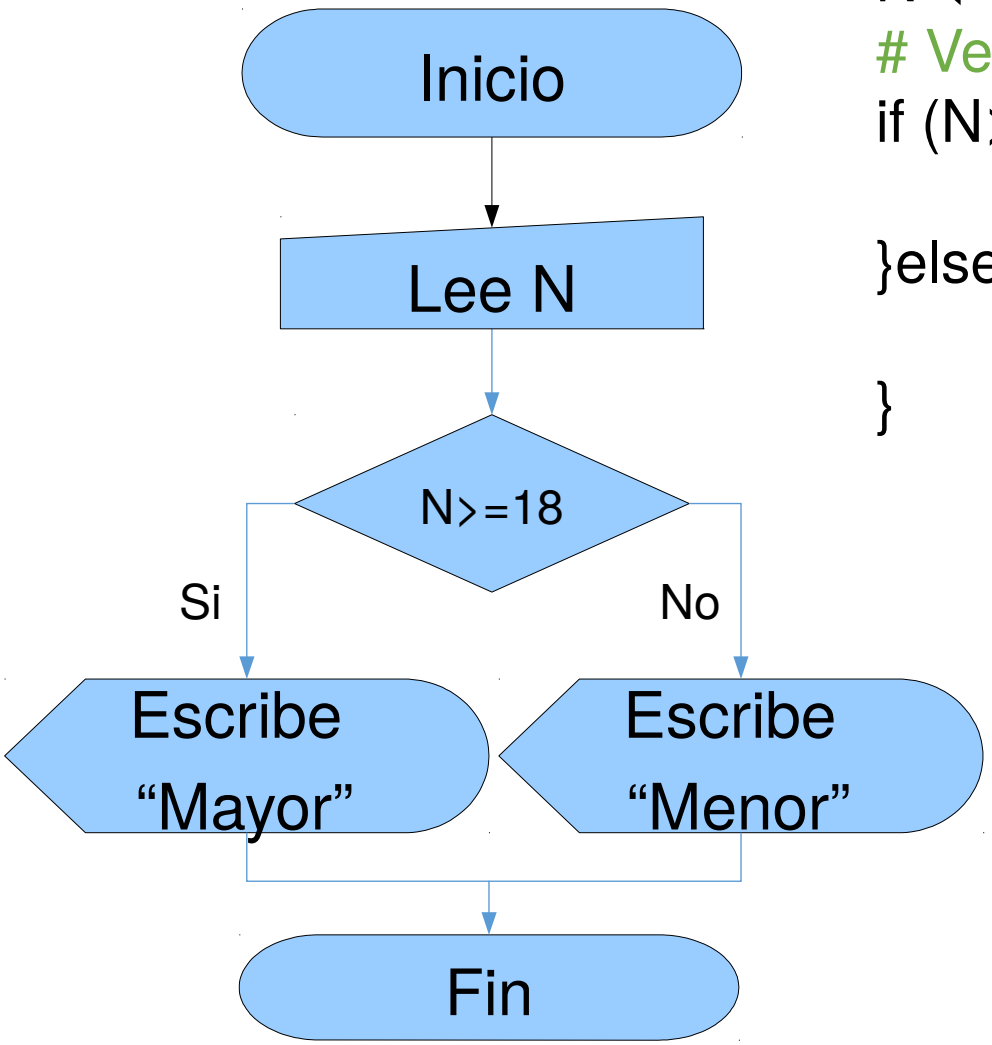
Hacer un programa que diga si un persona es mayor o menor de edad según un valor numérico:




```
N <- 20
# Vemos si es mayor o menor:
if (N>=18){
        print("Es mayor de edad")
}else{
        print("Es menor de edad")
}
```

Mayor de Edad

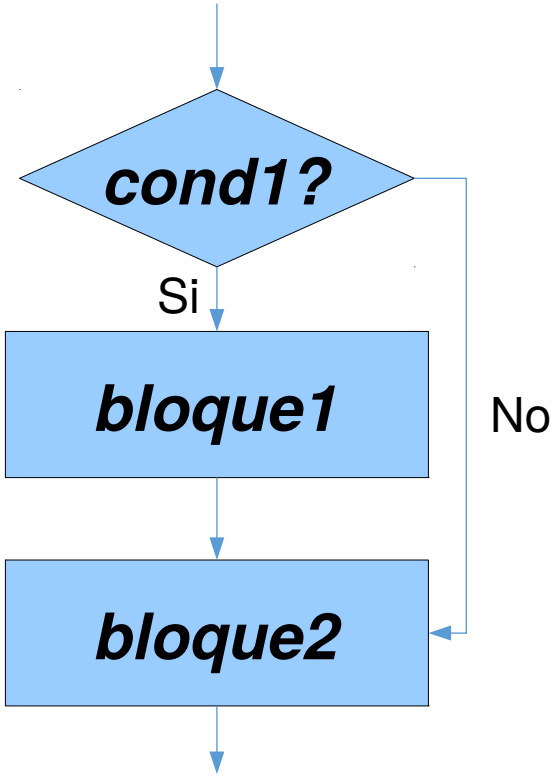
Hacer un programa que diga si un persona es mayor o menor de edad según un valor numérico:




```
N <- 20
# Vemos si es mayor o menor:
if (N>=18){
        print("Es mayor de edad")
}else{
        print("Es menor de edad")
}
```

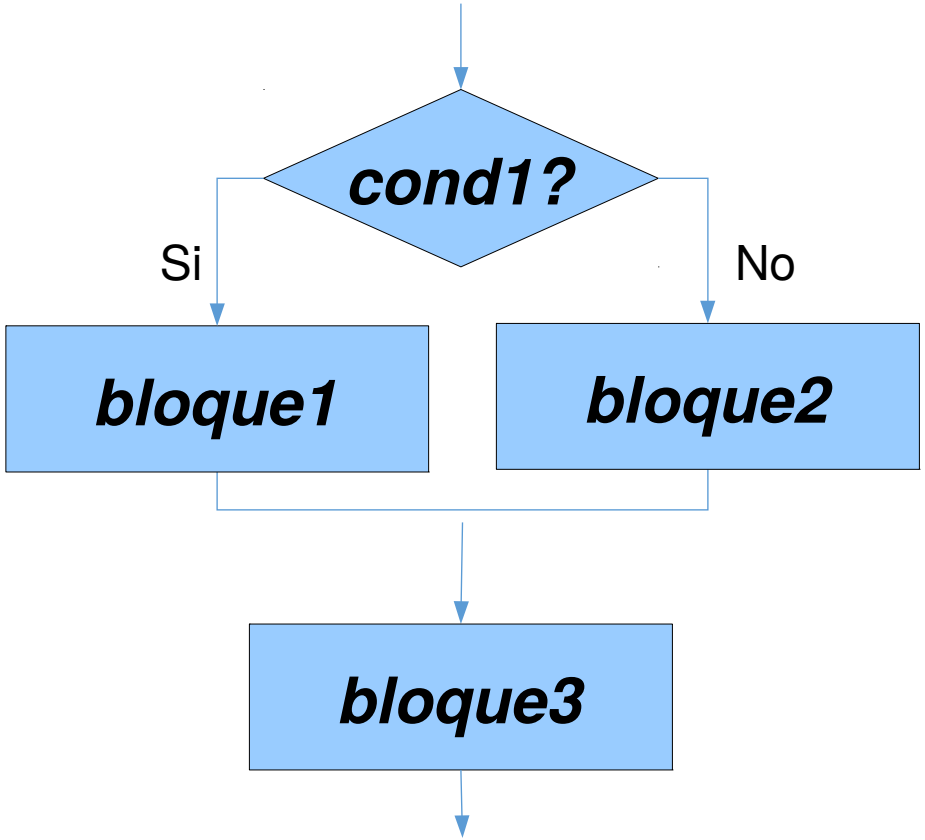
bifurcación/decisión/condición (no todas las sentencias se ejecutan dependiendo de condiciones que impongamos)

Sentencias de Control: Condicionales



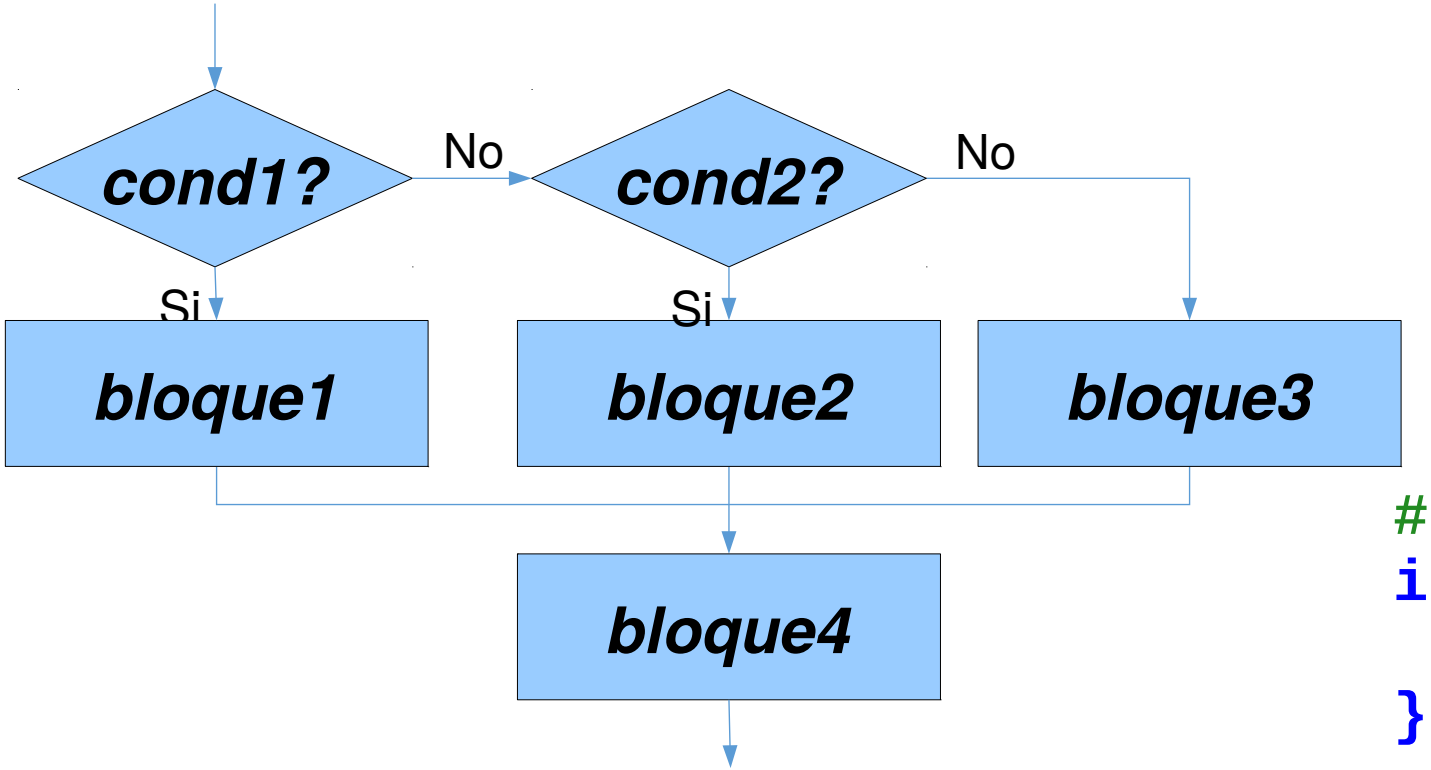

```r
# R
if (cond1){
  bloque1
}
bloque2
```

Sentencias de Control: Condicionales




```
# R
if (cond1){
  bloque1
}else{
  bloque2
}
bloque3
```

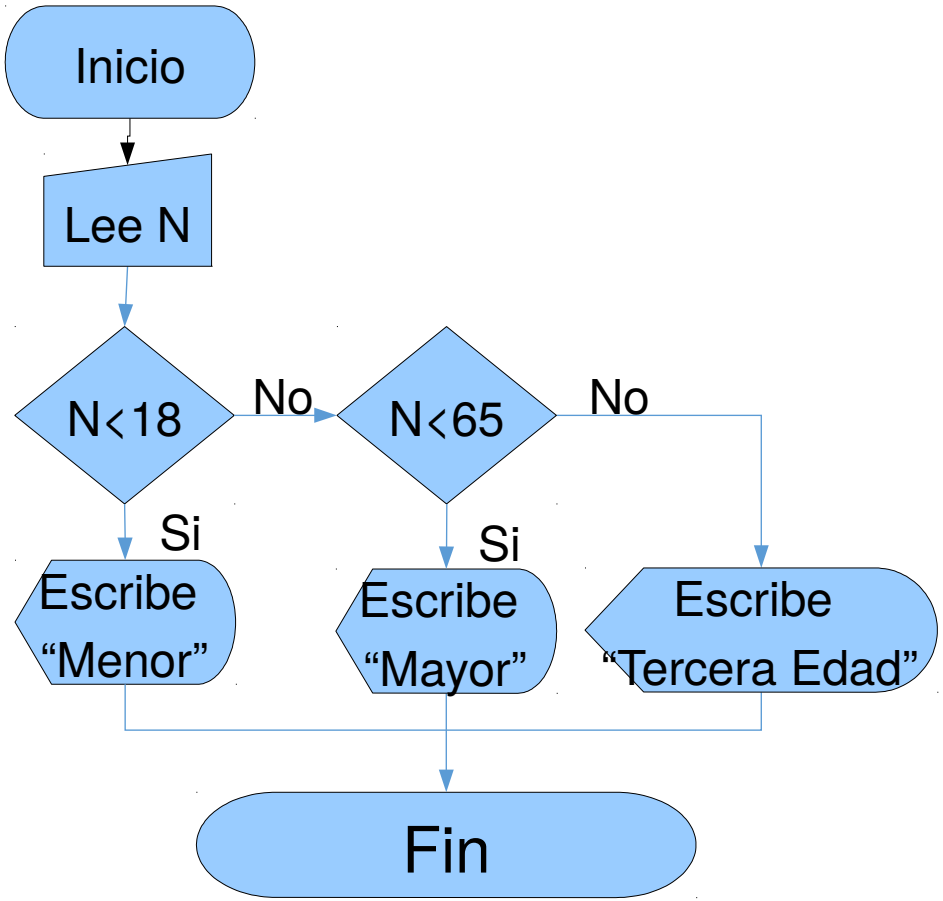
Sentencias de Control: Condicionales




```
# R
if (cond1){
  bloque1
}else if (cond2){
  bloque2
}else{
  bloque3
}
bloque4
```

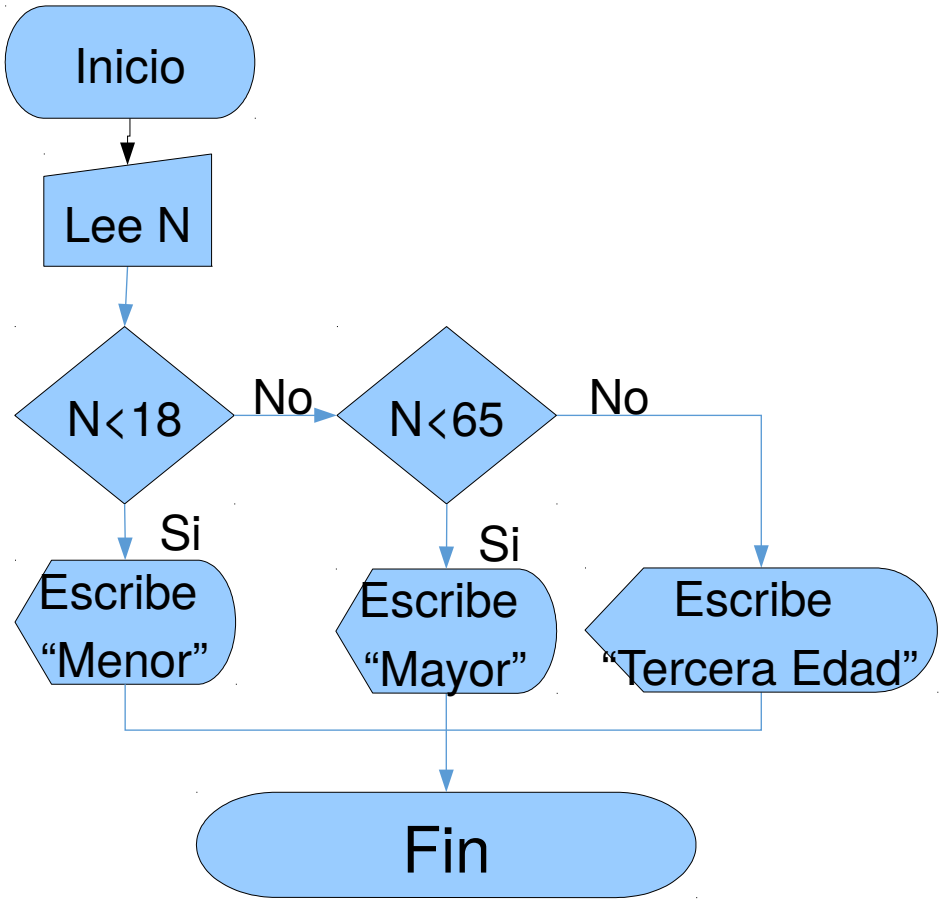
Menor, Mayor o Tercera Edad

Hacer un programa que solicite la edad y diga si es menor, mayor o de la tercera edad:



Menor, Mayor o Tercera Edad

Hacer un programa que solicite la edad y diga si es menor, mayor o de la tercera edad:




```
N <- 20
# Vemos si es mayor o menor:
if (N<18){
        print("Es menor de edad")
}else if (N<65){
        print("Es mayor de edad")
}else{
        print("Es de la tercera edad")
}
```

Operadores Básicos: Lógicos

Operación	Símbolo	Ejemplo
Y lógico (AND)	&	a & b
O Lógico (OR)	|	a | b
NO lógico (NOT)	!	!a

Los operadores lógicos nos permiten concatenar condiciones. Por ejemplo, la definición de un intervalo sería $0<X<18$, que es la conjunción de dos condiciones, $0<X$ y $X<18$, por lo que es necesario un operador lógico que permita realizar esta concatenación: $(0<X)$ & $(X<18)$.

Operadores Básicos: Lógicos

X	Y	Resultado (X & Y)
true (1)	true (1)	true (1)
true (1)	false (0)	false (0)
false (0)	true (1)	false (0)
false (0)	false (0)	false (0)

X	Y	Resultado (X | Y)
true (1)	true (1)	true (1)
true (1)	false (0)	true (1)
false (0)	true (1)	true (1)
false (0)	false (0)	false (0)

Operadores Básicos: Lógicos

X	Y	Resultado (X & Y)
true (1)	true (1)	true (1)
true (1)	false (0)	false (0)
false (0)	true (1)	false (0)
false (0)	false (0)	false (0)

X	Y	Resultado (X | Y)
true (1)	true (1)	true (1)
true (1)	false (0)	true (1)
false (0)	true (1)	true (1)
false (0)	false (0)	false (0)

X	Resultado (!X)
true (1)	false (0)
false (0)	true (1)

Operadores Básicos: Lógicos

X	Y	!(X & Y) = (!X | !Y)
true (1)	true (1)	false (0)
true (1)	false (0)	true (1)
false (0)	true (1)	true (1)
false (0)	false (0)	true (1)

X	Y	!(X | Y) = (!X & !Y)
true (1)	true (1)	false (0)
true (1)	false (0)	false (0)
false (0)	true (1)	false (0)
false (0)	false (0)	true (1)

Precedencia de operadores

() paréntesis el más interno es el que primero se ejecuta.

^ exponenciación.

* / \ multiplicación y división (igual precedencia).

+ - Suma y resta (igual precedencia).

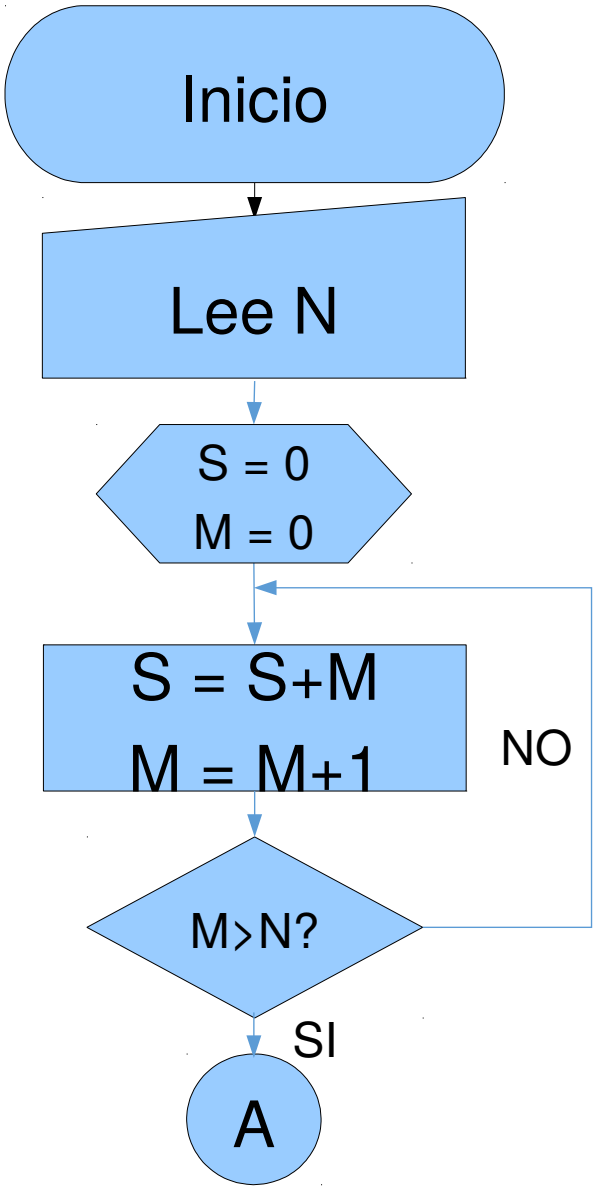
< <= > >= == ~= relacionales

& AND lógico

| OR lógico

Si dos o más operaciones tiene la misma precedencia, la expresión entonces será ejecutada de izquierda a derecha.

Sentencias de Control

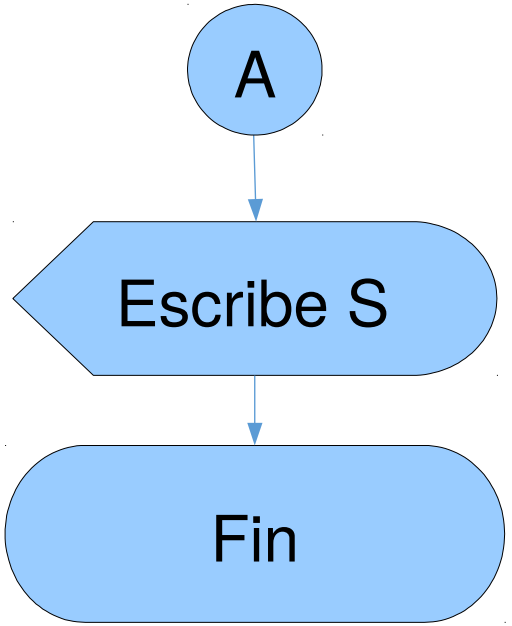


Iteración/Repetición/Ciclos:

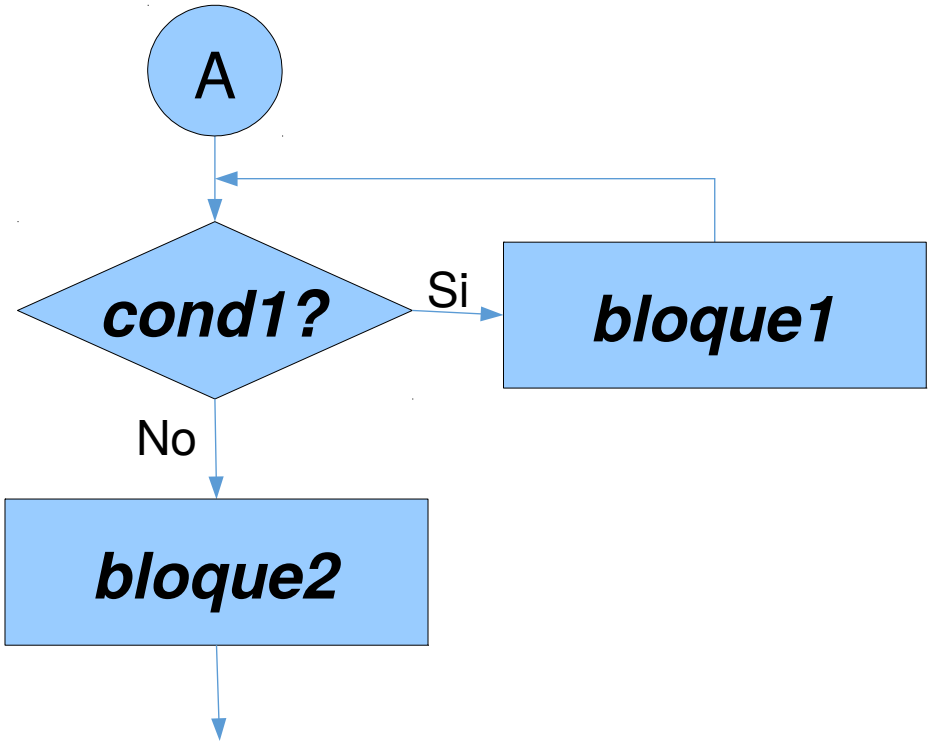
Los ciclos (bucles) permiten volver hacia atrás en el flujo de trabajo para repetir la ejecución de ciertas sentencias. Las órdenes básicas son:

while … (repite ciertas órdenes **mientras** se cumpla una condición)

for … (repite ciertas órdenes **para** unos valores dados)

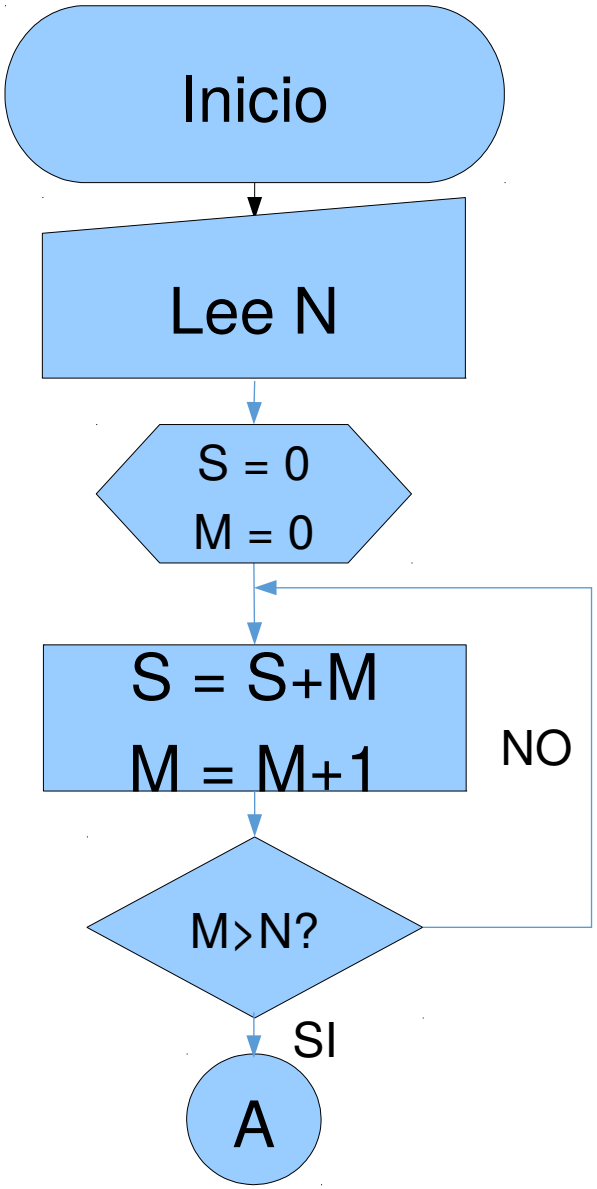


Sentencias de Control: Ciclos (Bucles)



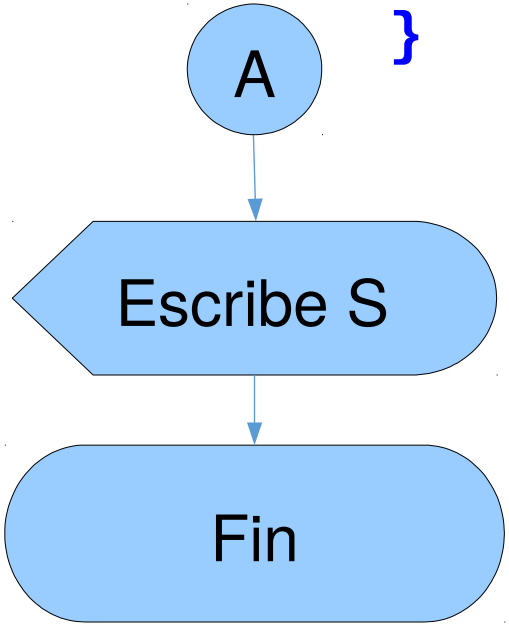

```r
# R
while (cond1){
   bloque1
}
bloque2
```

Sentencias de Control: Ciclos (Bucles)

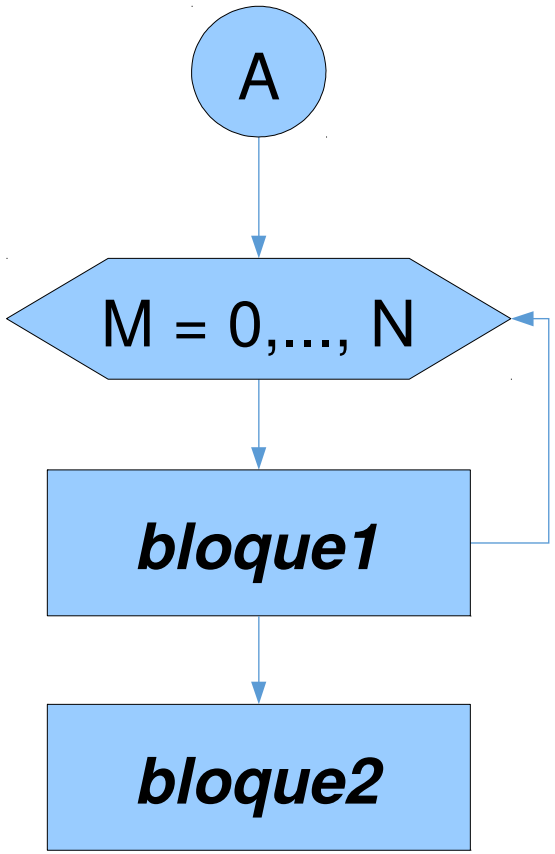


Hasta que el valor de **M** sea mayor que **N** el programa repetirá las operaciones actualizando **S** y **M**.

```
#R
while (M<=N){
    S ← S+M
    M ← M+1
}
```

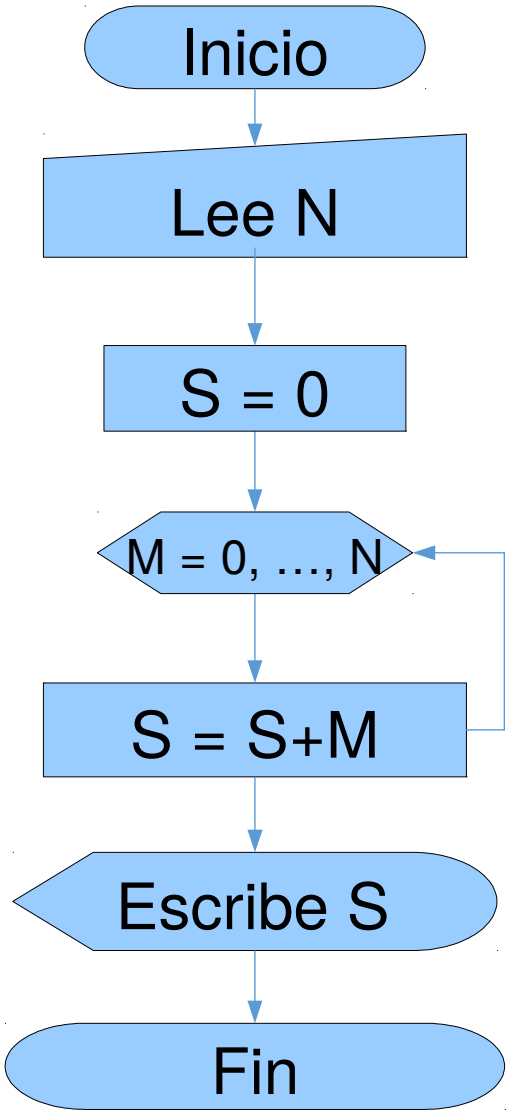


Sentencias de Control: Ciclos (Bucles)




```r
# R
for (M in c(0:N)){
  bloque1
}
bloque2
```

Sentencias de Control: Ciclos (Bucles)



Para los valores de **M**: 0, 1, 2, 3,..., **N**-1, **N** el programa sumará a **S** el valor de **M**.

```
# R
for (M in c(0:N)){
    S ← S+M
}
```

Ejercicios: Ciclos

Que calcule la raíz en el intervalo [0 2] del polinomio x^3-3x^2-2x+6 con una precisión de 1e-5 a través del método de la bisección.

Que calcule la raíz del polinomio anterior con el método de Newton-Raphson con la misma tolerancia tomando como condición inicial los extremos del intervalo [0 2]. ¿Qué soluciones obtienes? ¿Coinciden con las anteriores?

Resuelve numéricamente el sistema caótico de Lorenz.