



# Simplificando IFs anidados

---

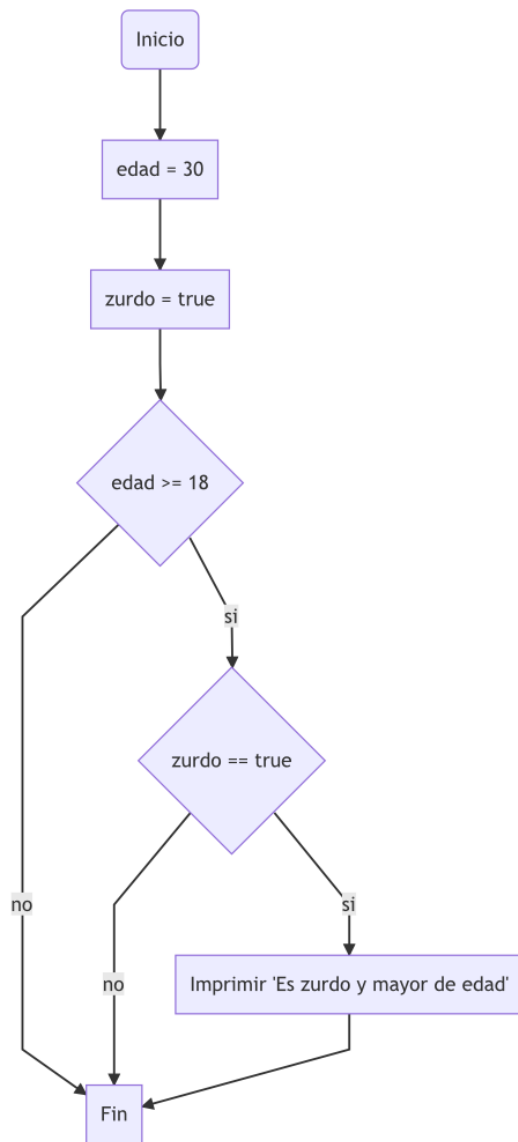
## Objetivos

- Simplificar problemas con condiciones anidadas en Ruby
- Entender complicaciones en ifs anidados en Ruby

## Identificando ifs anidados

---

Analicemos el siguiente ejemplo:



```
edad = 30
zurdo = true

if edad >= 18
  if zurdo == true
    puts 'Es zurdo y mayor de edad'
  end
end
```

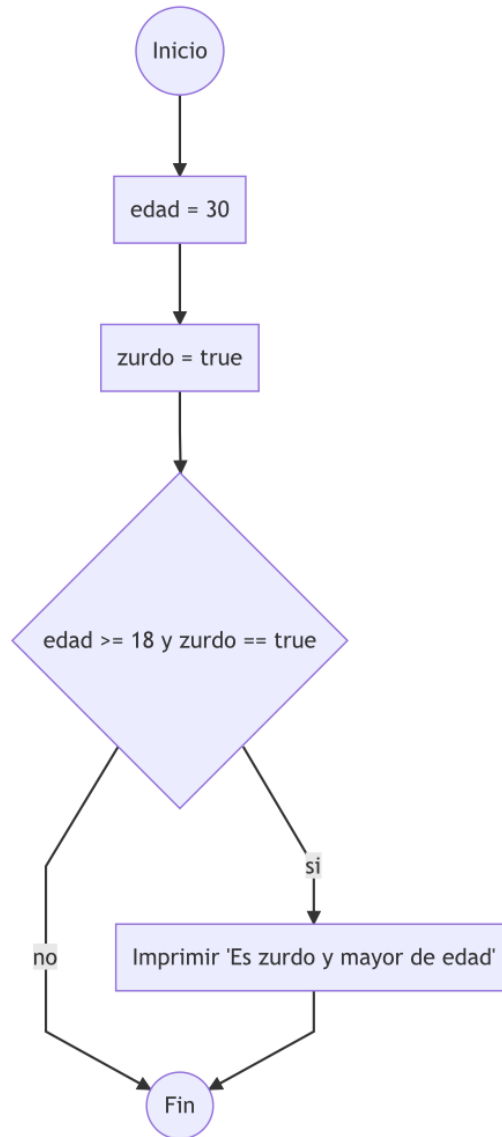
En el ejemplo anterior vemos un `if` dentro de otro `if`: a esto se le llama tener `if` anidados.

El código escrito muestra el texto sólo si se cumplen las dos condiciones.

# Operadores lógicos

Podemos simplificar el código para evaluar ambas condiciones en una misma instrucción `if`. Para una correcta utilización de instrucciones condicionales múltiples, debemos recordar la tabla de operadores lógicos:

¿Qué operador podemos utilizar si necesitamos que **ambas condiciones sean verdaderas**?



```
edad = 30
zurdo = true

if edad >= 18 && zurdo == true
  puts 'Es mayor de edad y zurdo'
end
```

Simplificando condiciones compuestas, el diagrama se simplifica bastante y expresa mejor el flujo del programa. Además, el código resulta mucho más fácil de comprender:

# Ejercicio de integración

---

Se busca crear un programa que solicite al usuario ingresar tres números. El programa debe determinar el mayor de ellos. Se asume que los números ingresados serán distintos.

¿Cómo lo podemos resolver?

Si bien es posible utilizar ifs anidados, vamos a utilizar lo aprendido para resolverlo de forma inteligente:

El **número 1** es mayor o igual que el **número 2** y mayor o igual que el **número 3**

```
puts 'Ingresa primer número: '
a = gets.to_i

puts 'Ingresa segundo número: '
b = gets.to_i

puts 'Ingresa tercer número: '
c = gets.to_i

if a >= b && a >= c
  puts "a es el mayor"
elsif b >= c
  puts "b es el mayor"
else
  puts "c es el mayor"
end
```