

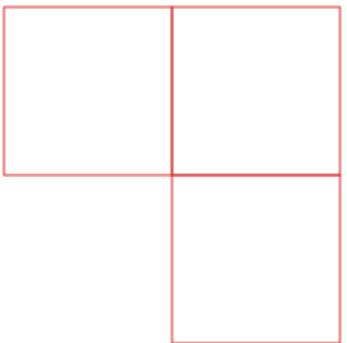
2

Sentencias de decisión



Ve más allá





CONTENIDOS

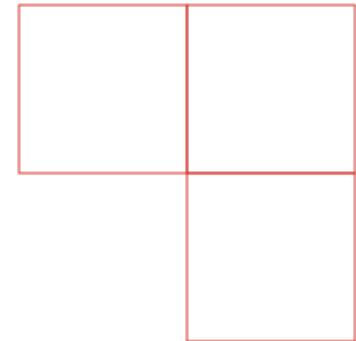
■ Unidad 2: Sentencias de Control

1. Expresiones condicionales
2. **Sentencias de decisión**
3. Sentencias de repetición



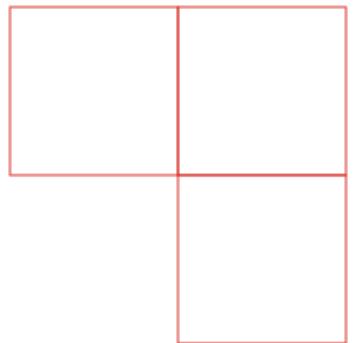


ÍNDICE



- Sentencias de decisión
 - 1. Introducción
 - 2. Sentencia if
 - Simple
 - Compuesta
 - Anidada
 - 3. Errores comunes
 - Sintaxis
 - Números decimales
 - Strings
 - Null
 - 4. Condiciones complejas
 - 5. Operador Ternario
 - 6. Sentencia Switch



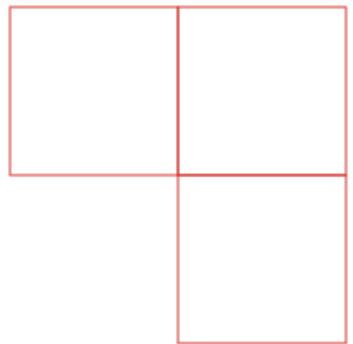


Introducción

Cuando necesitamos que la ejecución de una instrucción se realice sólo si se cumplen una serie de condiciones, lo que necesitamos son sentencias de decisión.

Vamos a conocer las distintas sentencias que tiene Java de este tipo, poniendo en práctica todo lo aprendido anteriormente sobre las expresiones condicionales.

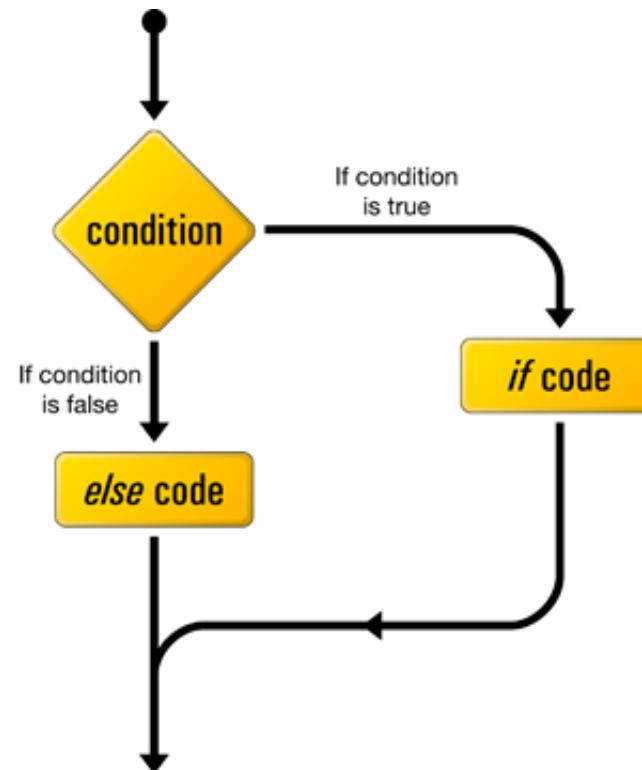


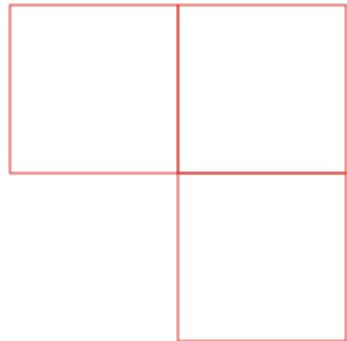


Sentencia if (simple)

Esta sentencia evalúa una expresión booleana y dependiendo de su valor true o false, ejecuta o no el bloque de sentencias asociado.

El diagrama de flujo de esta instrucción es el siguiente:





Ejercicio

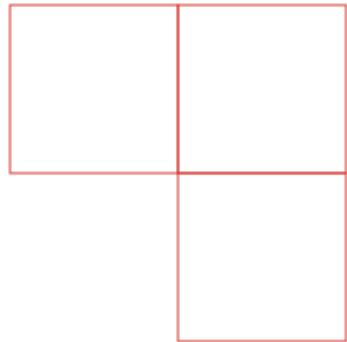
Imagina que estás haciendo una aplicación para una entidad bancaria y quieres avisar al usuario si está en números rojos. El programa, pedirá al usuario el saldo inicial y cuánto dinero quiere sacar. Tras lo anterior, se informará del saldo y sólo en caso de que dicho saldo sea negativo, se imprimirá un mensaje de aviso.

```
public static void main(String[] args) {  
    Scanner sc = new Scanner (System.in);  
  
    System.out.print("Introduce saldo inicial: ");  
    double saldo = sc.nextDouble();  
    System.out.print("Introduce importe a retirar: ");  
    double cantidad = sc.nextDouble();  
    System.out.println("Saldo final: " + (saldo - cantidad));  
    if (condicion) {  
        System.out.println("Estás en números rojos");  
    }  
}
```

¿Qué condición pondrías?

- a. saldo – cantidad > 0
- b. saldo – cantidad < 0
- c. saldo – cantidad >= 0
- d. saldo – cantidad <= 0

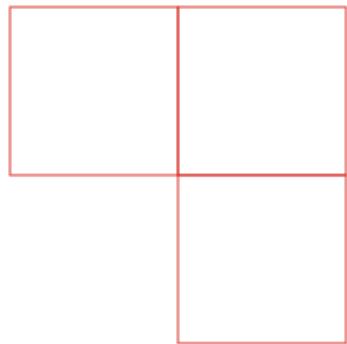
■
Programación/Sentencias de
decisión



Sentencias if (compuesta)

Decisions

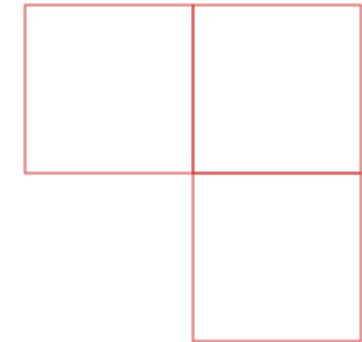




Sentencias if (compuesta)

```
public static void main(String[] args)
{
    Scanner in = new Scanner(System.in);
    System.out.print("Floor: ");
    int floor = in.nextInt();
    int actualFloor;

    // TODO: Modify the code to calculate the actual floor in
    // a building with neither a 13 or 14 floor
    if (floor > 13)
    {
        actualFloor = floor - 1;
    }
    else
    {
        actualFloor = floor;
    }
    System.out.println("Actual floor: " + actualFloor);
}
```



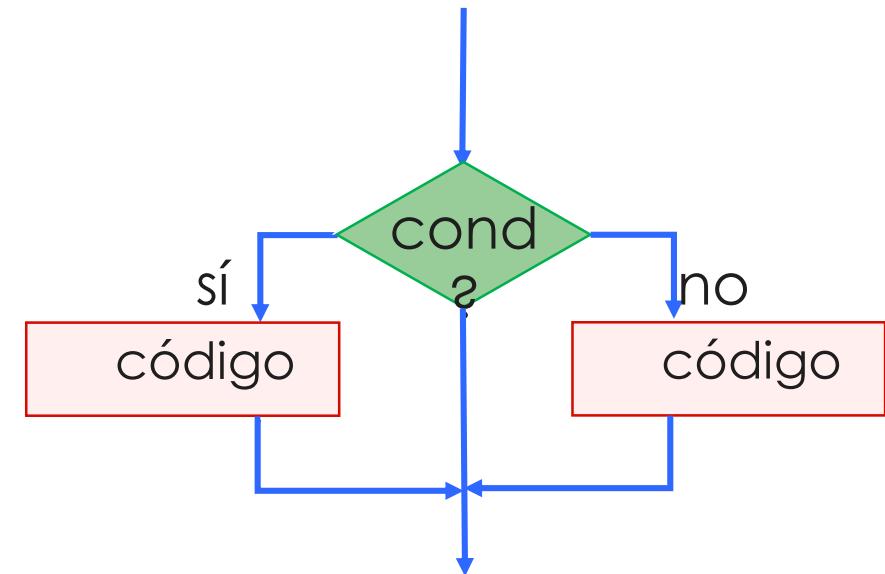
Sentencias if (compuesta)

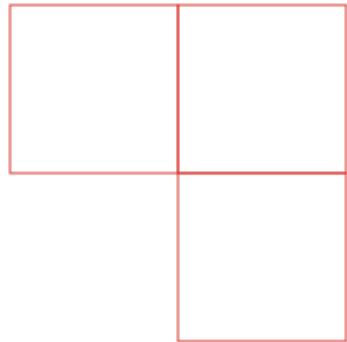
```
if (<boolean expresion>)
    <sentence>;
else
    <sentence>;
if (saldo>importe)
{
    saldo-=importe;
    System.out.println("Operación Realizada");
}
else
    System.out.printf("No se ha podido realizar la operación "+
        "para importe %d\n",importe);
System.out.printf("Saldo: %d\n",saldo);
```

Evalúa la condición (valor booleano).

Si la condición es cierta ejecuta la 1^a rama

Si es falsa ejecuta la 2^a rama.





Ejercicio

Queremos hacer un programa que solicite un número y muestre por pantalla si es par o no lo es.

```
public static void main(String[] args) {  
    Scanner sc = new Scanner(System.in);  
    System.out.print("Introduce un número: ");  
    int numero = sc.nextInt();  
    if (condicion)  
        System.out.println("El número no es par");  
    else  
        System.out.println("El número es par");  
}
```

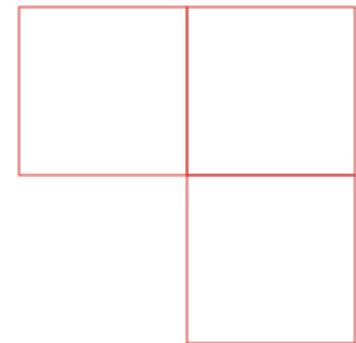
¿Qué condición pondrías?

- a. $\text{numero} \% 2 == 0$
- b. $!(\text{numero} \% 2 != 0)$
- c. $\text{numero} \% 2 != 0$
- d. $\text{numero} / 2 != 0$

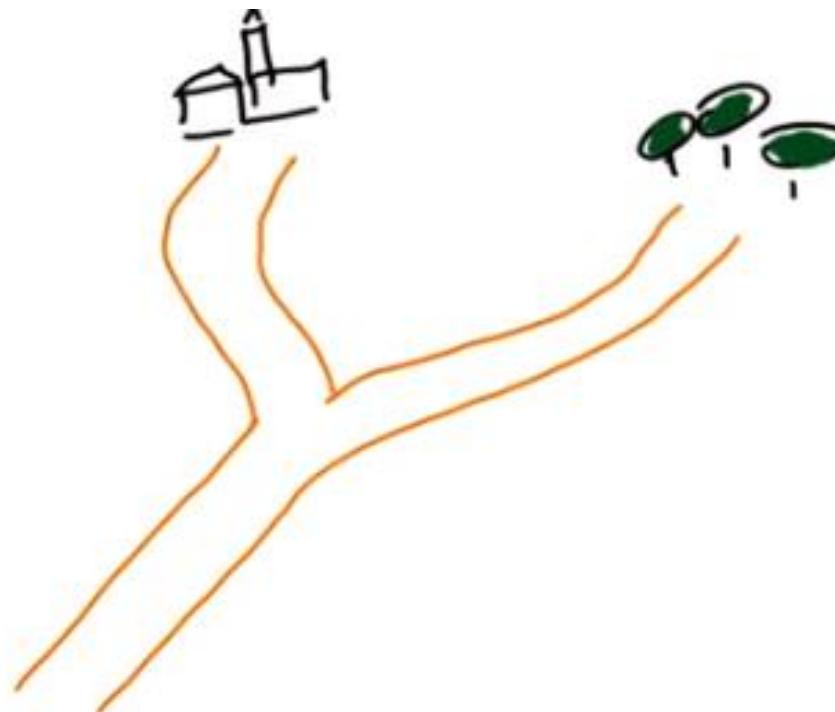


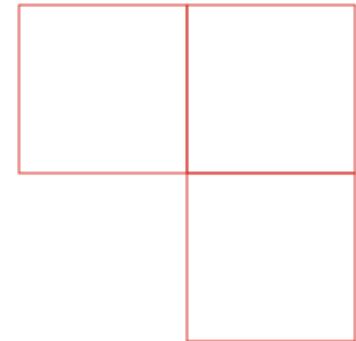
Sentencias if (compuesta)

Programación/Sentencias de
decisión



Decisions





Sentencias if (compuesta)

```
if (originalPrice > 100)
{
    discountedPrice = originalPrice - 20;
}
else
{
    discountedPrice = originalPrice - 10;
}
```



Flight to Las Vegas, original price: \$95



Flight to Chicago, original price: \$100



Flight to New York City, original price: \$105



<- solución

```
if (originalPrice > 100)           if (originalPrice < 100)
{
    discountedPrice = originalPrice - 20;   {
                                                discountedPrice = originalPrice - 10;
}
else                                     else
{
    discountedPrice = originalPrice - 10;   {
                                                discountedPrice = originalPrice - 20;
}
}
```



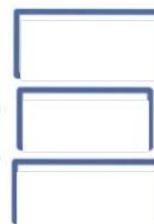
85 Flight to Las Vegas, original price: \$95



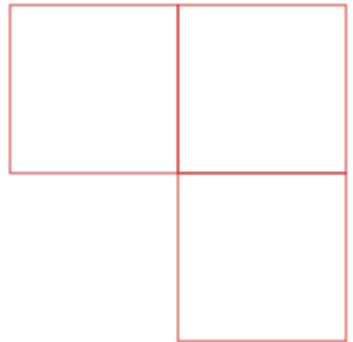
90 Flight to Chicago, original price: \$100



85 Flight to New York City, original price: \$105



<- solución



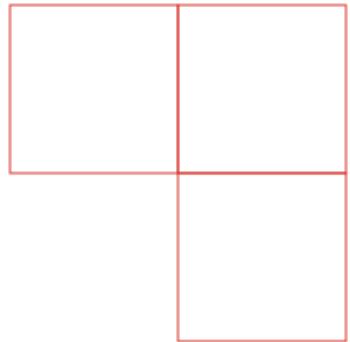
Sentencias if (Anidada)

Las instrucciones if se pueden anidar, es decir, que un if esté dentro de otro if.

Se recomienda que esta concatenación se realice por la rama else, ya que se leería: “sino si...”.

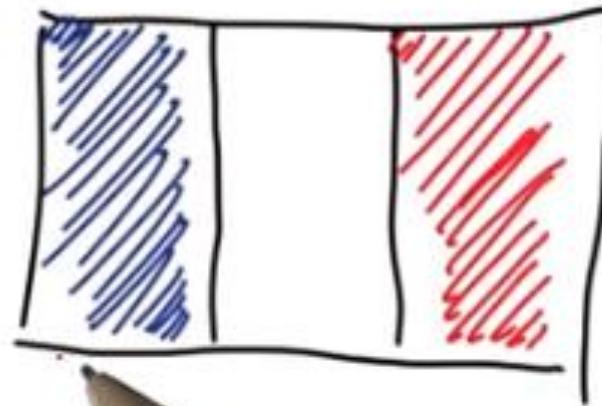
Se suelen utilizar para decisiones más complejas, donde hay más de dos alternativas. Es decir, permite seleccionar entre tres o más opciones posibles.

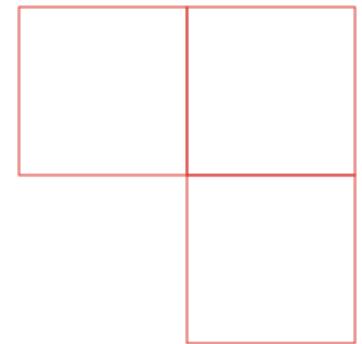
```
if(saldo>=importe)
    realizar_operacion();
else if (credito)
    realizar_operacion();
else
    cancelar_operación();
```



Sentencias if (Anidada)

if/else if/else





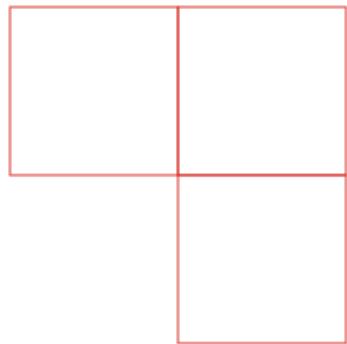
Ejercicio

Realizar un programa que solicite un número entero positivo y muestre por pantalla si tiene 1, 2 o más cifras.

```
public static void main(String[] args) {  
    Scanner in = new Scanner (System.in);  
    System.out.print("Introduzca un número: ");  
    int numero = in.nextInt();  
  
    if (condicion1) {  
        System.out.println("1 cifra");  
    } else if (condicion2) {  
        System.out.println("2 cifras");  
    } else {  
        System.out.println("más de 2 cifras");  
    }  
}
```

¿Qué condición pondrías en cond1 y cond2?

- numero < 10 y numero < 100
- numero < 100 y numero < 10
- numero <= 10 y numero <= 100
- numero < 10 y (numero >= 10 && numero < 100)



Errores comunes (Sintaxis)

1. La condición dentro de la declaración If no evalúa un valor booleano.

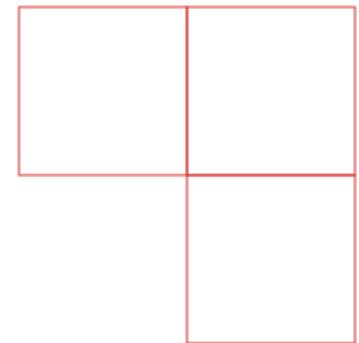
Por ejemplo:

```
//WRONG
int number = 0;
if (number) {           //some statements here
}
```

La variable number no es un tipo booleano

2. Escribir elseif en vez de else if.





Errores comunes (Números decimales)

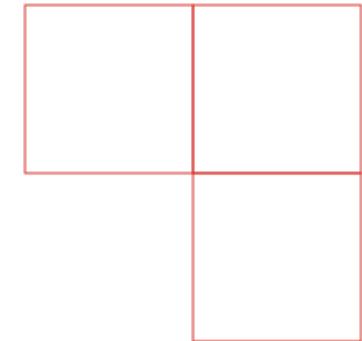
Comparing Decimal Numbers



```
public class EqualNumbers
{
    public static void main(String[] args)
    {
        double original = 2;
        double root = Math.sqrt(original);
        double rootSquared = root * root;
        if (rootSquared == original)
        {
            System.out.println("They are the same");
        }
        else
        {
            System.out.println("rootSquared is "
                + rootSquared);
        }
    }
}
```



<- solución

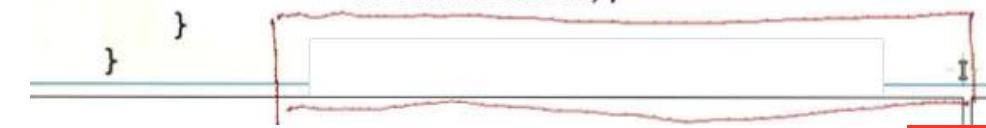


Errores comunes (Strings)

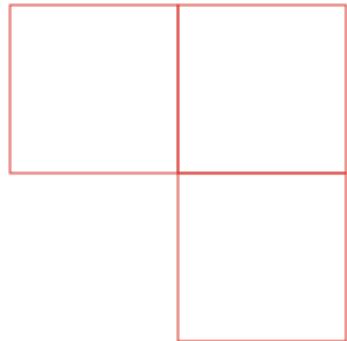
Comparing Strings



```
public class EqualStrings
{
    public static void main(String[] args)
    {
        String first = "Uda";
        String second = "city";
        String firstAndSecond = first + second;
        String third = "Udacity";
        if (firstAndSecond == third)
        {
            System.out.println("They are the same");
        }
        else
        {
            System.out.println("firstAndSecond is "
                + firstAndSecond);
        }
    }
}
```



<- solución



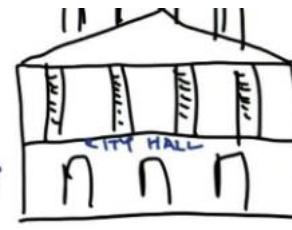
Errores comunes (Null)

null



String statement = ...;

Did the mayor say "No comment"?

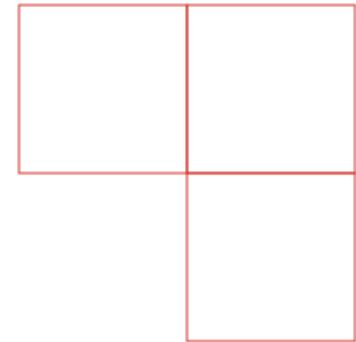


Did the mayor say nothing at all?

Did the reporter not get a statement?



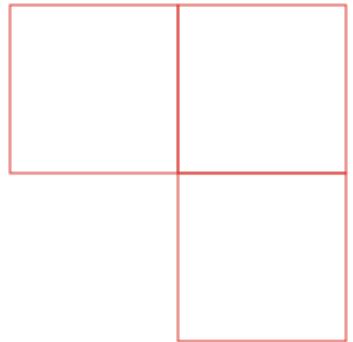
<- solución



Condiciones complejas

And/Or/Not





Condiciones complejas

Suppose that x and y are two die values.

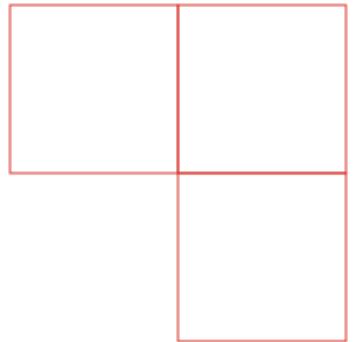
How would you check whether both of them are 3?

if() Write a condition here.

How would you check whether at least one of them is 3?

if() Write a condition here.

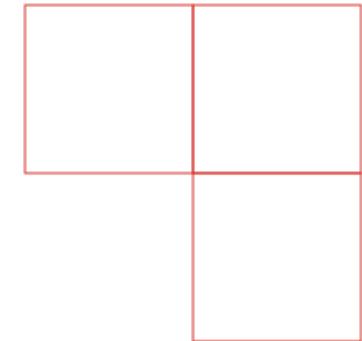




Condiciones complejas

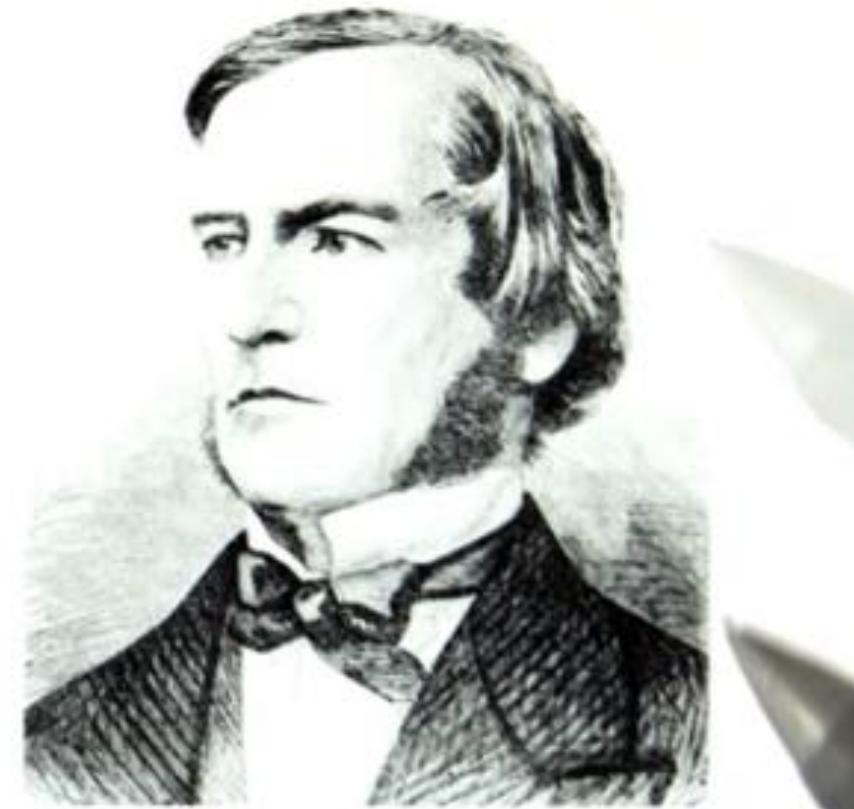
```
public class DiceGame
{
    /**
     * Calculates the score of a round of this game.
     * x the first roll
     * y the second roll
     * @return a score of 1 if exactly one is 3, a score of 0 otherwise
     */
    public int exactlyOneThree(int x, int y)
    {
        if (...) // TODO: if exactly one of x and y is 3
        {
            return 1;
        }
        else
        {
            return 0;
        }
    }
}
```



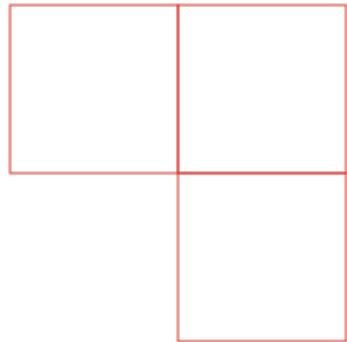


Condiciones complejas

Boolean Logic



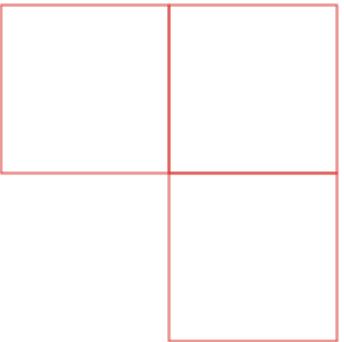
George Boole



Condiciones complejas

Input Validation



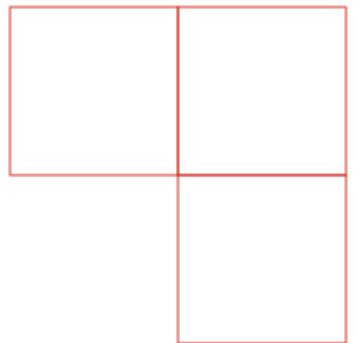


Condiciones complejas

```
import java.util.Scanner;
/*
    This program simulates an elevator panel that skips the 13th floor, checking for
    input errors.
*/
public class ElevatorDemo
{
    public static void main(String[] args)
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Floor: ");
        if (...) //check if the input is an integer. Hint: use the Scanner method hasNextInt.
        {
            // Now we know that the user entered an integer
            int floor = in.nextInt();
            if (...)
            {
                System.out.println("Error: There is no thirteenth floor.");
            }
            else if (...)
            {
                System.out.println("Error: The floor must be between 1 and 18.");
            }
            else
            {
                // Now we know that the input is valid

                int actualFloor = floor;
                if (...)
                {
                    actualFloor = floor - 1;
                }
                System.out.println("The elevator will travel to the actual floor "
                    + actualFloor);
            }
        }
        else
        {
            System.out.println("Error: Not an integer.");
        }
    }
}
```





Operador ternario

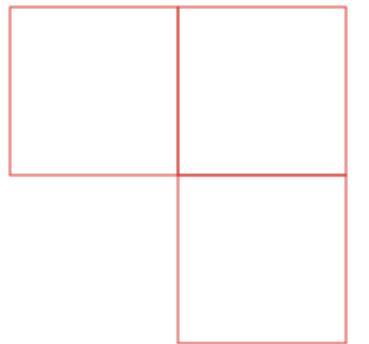
La sintaxis es la siguiente:

```
<condición>?<expresión1>:<expresión2>
```

El resultado es expresión1 si la condición es cierta y expresión2 si no

```
int a = 5 > 6 ? 3 : 4;
```

Da como resultado que a = 4, ya que la expresión 5>6 es false.



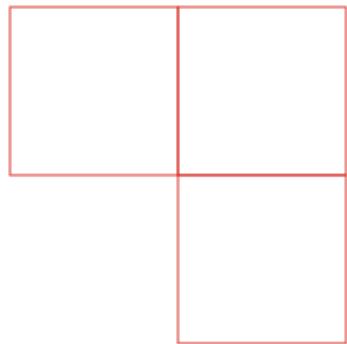
Ejercicio

Realizar un programa que solicite dos números y muestre por pantalla el mayor de ellos, utilizando el operador ternario.

```
public static void main(String[] args) {  
    Scanner in = new Scanner (System.in);  
    System.out.print("Introduzca el primer número: ");  
    int num1 = in.nextInt();  
    System.out.print("Introduzca el segundo número: ");  
    int num2 = in.nextInt();  
  
    int mayor = ...;  
  
    System.out.println("El mayor de los números es " + mayor);  
}
```

¿Cómo lo completarías?

- a. num2 > num1 ? num1 : num2
- b. num1 > num2 ? num1 : num2
- c. num1 ? num2 > num1 : num2
- d. num1 : num2 > num1 ? num2



Sentencia switch

Se usa cuando se abren múltiples opciones

- como alternativa al uso de un if anidado
- si los valores a comparar son de tipo byte, char, short, int o String
- si los valores con que se compara son conocidos a priori
 - constantes o literales

```
switch (expresión) {  
    case constante1:  
        sentencia1;  
        ...  
        break;  
    case constanteN:  
        sentenciaN;  
        ...  
        break;  
    default:  
        sentencia;  
        ...  
}
```

Evalúa la expresión

Si encuentra un **case** con el mismo valor

ejecuta las sentencias que le siguen

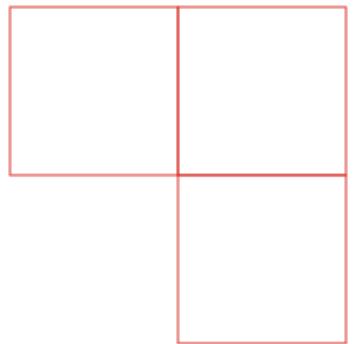
Si no encuentra ninguna constante que coincida

busca la línea **default** (opcional).

La sentencia **break** finaliza la ejecución de una rama.

Si no se pone continuará ejecutando instrucciones.

(de los siguientes **case**)



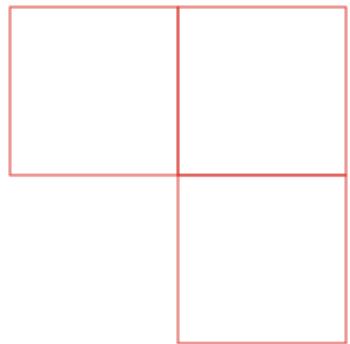
Ejercicio

Escribe un programa que solicite la introducción de un número entre 1 y 3 y escriba por pantalla si es “uno”, “dos” o “tres”.

```
public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    System.out.print("Introduzca 1, 2 ó 3: ");
    int numero = in.nextInt();
    String texto = "";
    switch (...) {
        case 1:
            texto = "Uno";
            break;
        case 2:
            texto = "Dos";
            break;
        case 3:
            texto = "Tres";
            break;
        default:
            texto= "...era 1, 2 ó 3";
    }
    System.out.println(texto);
}
```

¿Cómo lo completarías?

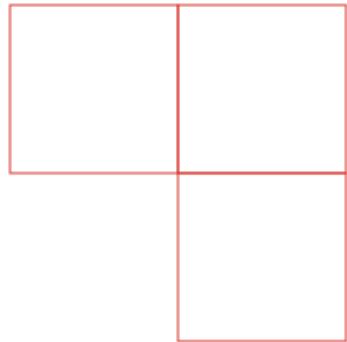
- a) texto
- b) numero==1
- c) numero
- d) 1,2,3



Ejercicio 2

Escribe un programa que solicite la introducción de un número entre 1 y 4 y escriba por pantalla si es “uno o dos”, “tres” o “cuatro”.

```
public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    System.out.print("Introduzca 1, 2, 3 ó 4: ");
    int numero = in.nextInt();
    String texto = "";
    switch (numero) {
        case 1:
        case 2:
            texto = "Uno o Dos";
            break;
        case 3:
            texto = "Tres";
            break;
        case 4:
            texto = "Cuatro";
            break;
        default:
            texto= "...era 1, 2, 3 ó 4";
    }
    System.out.println(texto);
}
```



Consejo final



APUNTE...

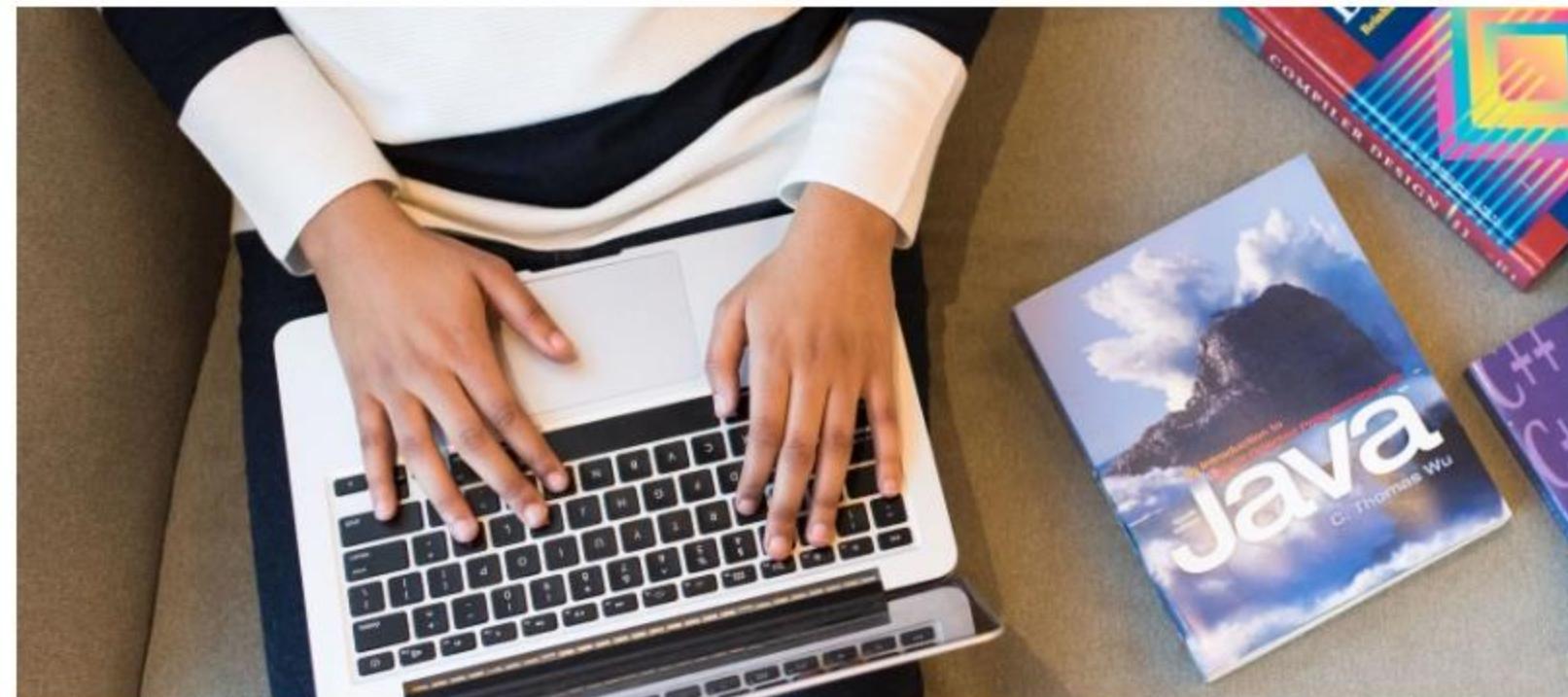
Los 10 mejores blogs de Java para programadores de todos los niveles



By Editorial de Geekflare in Carrera de agosto 10, 2022



Geekflare cuenta con el apoyo de nuestra audiencia. Podemos ganar comisiones de afiliados comprando enlaces en este sitio.





**Universidad
Europea**

GRACIAS

Pedro J. Camacho

Universidadeuropea.com

Ve más allá