

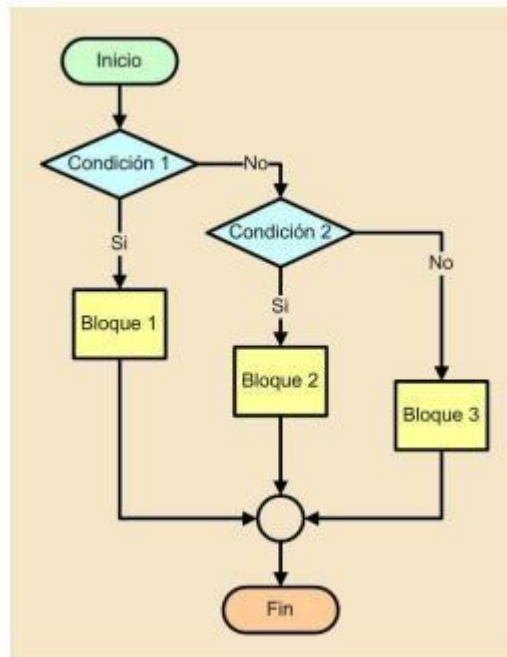
Sintaxis en Java: Condicional

1. Sentencias Condicionales

1.1 Introducción

Luego de haber visto cómo definimos variables, ahora tenemos como objetivo principal conocer cómo utilizarlas en nuestro código. La primera sentencia que veremos es la sentencia **condicional**. Esta, implica la decisión de qué bloque de código se debe ejecutar según el flujo de datos.

La idea de esta sentencia, puede ilustrarse en el siguiente diagrama:



Dado una condición, que puede ser verdadera o falsa, se decide qué bloque de código ejecutar. Dentro de los bloques que se muestran en la imagen, a su vez, podemos encontrarnos con una nueva sentencia que deba decidir (condicionales anidados), y así sucesivamente.

Cuando vimos pseudocódigo representábamos el condicional con “SI” y “SINO”, en **Java la sentencia es “if” y “else” (en minúscula).**

Veamos un ejemplo de un código Java:

```
Programa.java x
1 package primerejemplo;
2
3 no usages new *
4 public class Programa {
5
6     no usages new *
7     public static void main(String[] args) {
8
9         int numero1 = 50;
10        int numero2=35;
11
12        if(numero1>=numero2){
13
14            //En este momento de la ejecución, esta se dará cuando se cumpla la condición numero1>=numero2
15            System.out.println("La condición es verdadera!");
16        }
17    }
18 }
```

- En este caso, tenemos una condición de comparación entre dos números, y en base a ella se decide que ejecutar.
- ¿Que evaluará la condición en este caso? true. Como consecuencia de eso, se pasa a ejecutar el código que se encuentra dentro del bloque if, que se encuentra comprendido entre las llaves que lo preceden.
- En este caso, únicamente se ejecutará el print que vemos allí. Luego, “salimos” del bloque del if, y al no haber más código, terminará la ejecución de nuestro programa.
- Puede ocurrir que luego de los bloques de if haya más código a ejecutar y al finalizar dicho bloque se pasa a la ejecución del código posterior.

1.2 Sintaxis de if

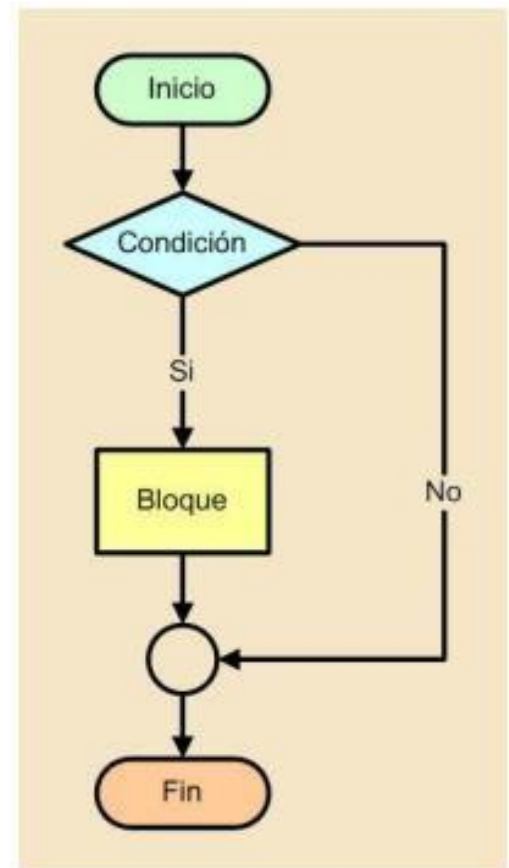
1.2.1 VARIANTE 1: Sentencia IF Simple

La primer variante (y la más simple) es la que se mostró en el ejemplo anterior. Una simple decisión, que determina si cierto código se ejecuta o no. La sintaxis es la siguiente:

```
if (XXX){  
  
}
```

Donde:

- XXX es una condición que devuelve un valor booleano: true o false (para decidir si se cumple o no).
- Esta puede ser una variable de tipo booleano, o una expresión que retorne un valor de este tipo (ej. $3 \neq 4$ devuelve true).
- Dentro de las llaves, debemos colocar el código que queremos ejecutar en caso de que la condición evalúe como true.



Veamos algunos ejemplos:

Ejemplo 1

```
Ejemplo1.java x
1 package ejs2condicionales;
2
3 no usages NicoDQ
4 public class Ejemplo1 {
5     no usages NicoDQ
6     public static void main(String[] args) {
7         int var1 = 50;
8         int var2 = 50;
9
10        String resultado = "Aún no entre al IF";
11
12        if(var1==var2){
13            resultado = "Cambio el valor de la variable resultado";
14        }
15
16        System.out.println(resultado);
17    }
18 }
19
20
21
```

Ejemplo 2

```
Ejemplo2.java x
1 package ejs2condicionales;
2
3 no usages NicoDQ
4 public class Ejemplo2 {
5     no usages NicoDQ
6     public static void main(String[] args) {
7         int var1 = 50;
8         int var2 = 50;
9
10        String resultado = "";
11
12        if(var1+var2==105){
13            resultado = "Cambio el valor de la variable resultado";
14        }
15
16        System.out.println(resultado);
17    }
18 }
19
20
21
```

Ejemplo 3

```
Ejemplo3.java x
1 package ejs2condicionales;
2
3 no usages NicoDQ
public class Ejemplo3 {
4
5 no usages NicoDQ
    public static void main(String[] args) {
6
7        int var1 = 50;
8        int var2 = 50;
9
10       boolean variableBoolean = var1 <= var2;
11
12       double resultado = 12.8;
13
14       if(variableBoolean){
15
16           int variableDentroDeIf = 20;
17           resultado = 13.56 + variableDentroDeIf;
18       }
19
20
21       System.out.println(resultado);
22   }
23
24 }
25
```

Observaciones:

- La expresión que se encuentra entre paréntesis de la sentencia if, debe ser de tipo booleano. Esto es una variable de este tipo, o alguna expresión que retorna un tipo booleano (comparaciones, igualdades, operaciones lógicas).
- Dentro de los bloques delimitados entre llaves, toda variable que definamos allí, será local a este bloque. Esto significa, que fuera del bloque, de definición, no va a ser reconocida. A esto llamamos “scope” o alcance de la variable.

Ejemplo 4

```
Ejemplo4.java x
1 package ejs2condicionales;
2
3 no usages  ± NicoDQ
4 public class Ejemplo4 {
5     //En este ejemplo vemos el alcance de la variable.
6     no usages  ± NicoDQ
7     public static void main(String[] args) {
8         if(true){
9             int i = 90;
10        }
11
12        int u = i;
13    }
14
15 }
16
17 }
18
19 }
```

En el ejemplo 4, vemos como el programa no compila si queremos utilizar una variable **declarada** dentro de un bloque de código ({}), por fuera de este.

Esto aplica no sólo con variables declaradas dentro del bloque de if, sino que dentro de cualquier bloque de código delimitado por {}.

Ejemplo 5

```
Ejemplo5.java x
1 package ej2condicionales;
2
3 public class Ejemplo5 {
4     public static void main(String[] args) {
5
6         int var1 = 50;
7         int var2 = 50;
8
9         boolean variableBoolean = var1 <= var2;
10
11         double resultado = 12.8;
12
13         if(variableBoolean){
14             int variableDentroDeIf = 20;
15
16             if(variableDentroDeIf != 67){
17                 System.out.println("Estamos dentro del segundo IF.");
18             }
19         }
20
21         System.out.println(resultado);
22
23     }
24 }
25
26 }
27
```

Como puede verse en el ejemplo, podemos anidar sentencias de if. Esto significa, que dentro del bloque de un if, puedo colocar otra de ser necesario.

1.2.2 VARIANTE 2: Sentencia IF – ELSE

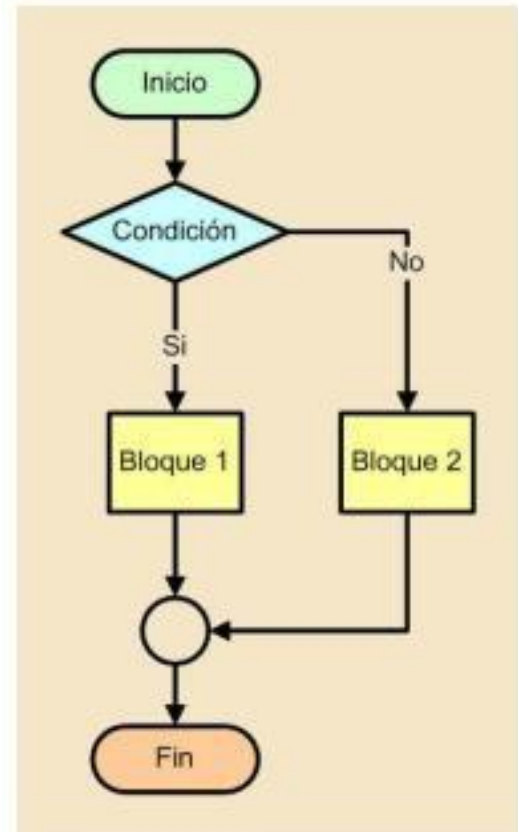
La segunda variante, trata de darnos una alternativa de código cuando la expresión que evaluemos no sea verdadera. De este modo expresamos algo del estilo “si pasa X entonces hacer J, de lo contrario hacer K”. Corresponde a lo que en pseudocódigo representamos como “SINO”. La sintaxis, es la siguiente:

if (XXX){

} else{

} Donde:

- XXX es una condición que devuelve un valor booleano: true o false (para decidir si se cumple o no).
- Esta puede ser una variable de tipo booleano, o simplemente una expresión que retorne un valor de este tipo.
- Dentro de las primeras llaves, debemos colocar el código que queremos que se ejecute si la condición es true, luego, dentro de las segundas llaves, el código alternativo en caso de que la condición sea false.



Veamos algunos ejemplos:

Ejemplo 1


```
Ejemplo1.java x
1 package ejs2condicionalesifelse;
2
3 public class Ejemplo1 {
4
5     public static void main(String[] args) {
6
7         int var1 = 50;
8         int var2 = 50;
9
10        String resultado = "Aún no entre al IF";
11
12        if(var1 == var2){
13            resultado = "Cambio el valor de la variable resultado, porque dio true";
14        }else{
15            resultado = "Cambio el valor de la variable, pero ahora porque dio false";
16        }
17
18        System.out.println(resultado);
19    }
20 }
21
22
```

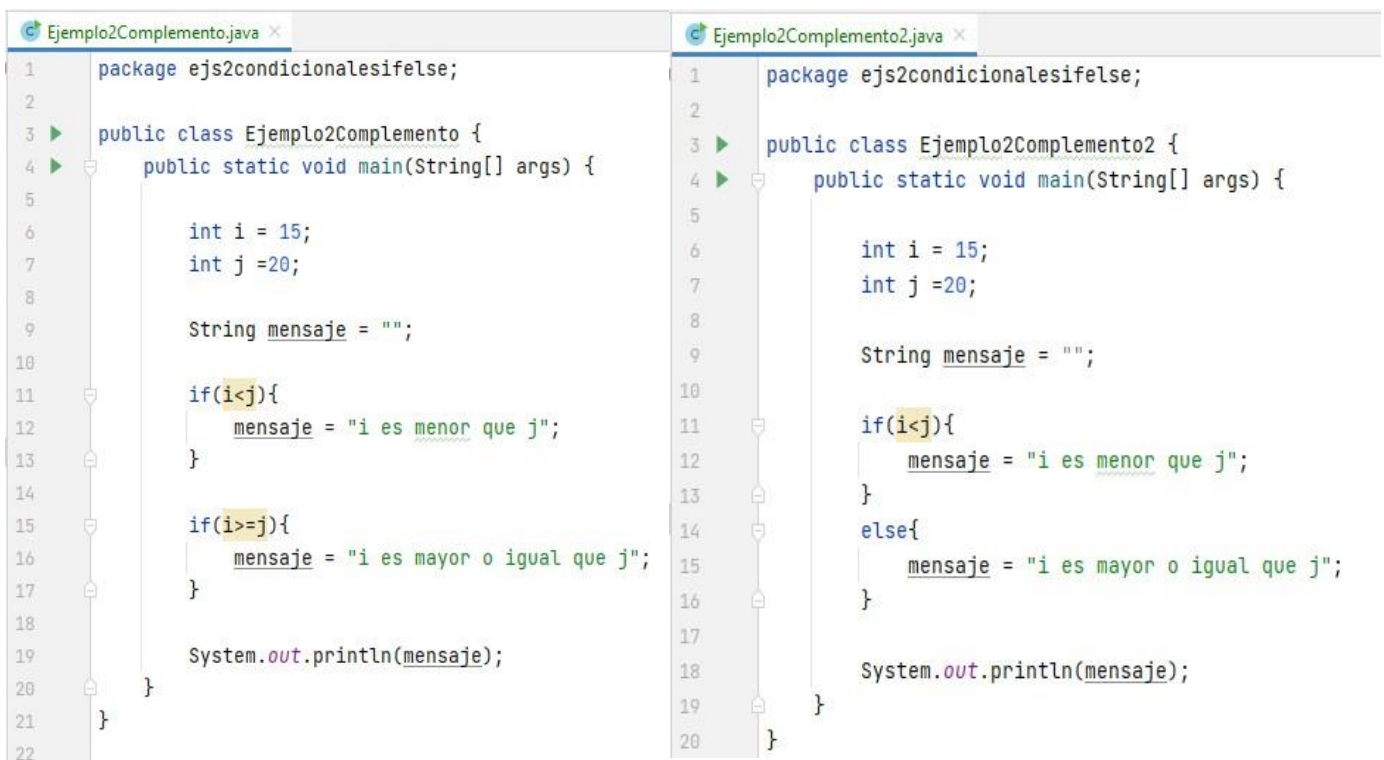
Ejemplo 2

```
Ejemplo2.java x
1 package ejs2condicionalesifelse;
2
3 public class Ejemplo2 {
4
5     public static void main(String[] args) {
6
7         int var1 = 50;
8         int var2 = 50;
9
10        String resultado = "Aún no entre al IF";
11
12        if(var1 == var2){
13            resultado = "Cambio el valor de la variable, porque dio true";
14        }else{
15            resultado = "Cambio el valor de la variable, pero ahora porque dio false";
16
17            boolean otroBooleano = var1!=var2;
18
19            if(otroBooleano){
20                resultado = "Cambiamos por tercera vez, pero llegamos a este código?";
21            }
22        }
23
24        System.out.println(resultado);
25    }
26 }
27
28
29
```

Observaciones:

➤ Como podemos ver en el segundo ejemplo, podemos anidar dentro de cada bloque (if o else), otras sentencias if.

Por ejemplo, estos dos códigos ejecutan lo mismo:



```
1 package ejs2condicionalesifelse;
2
3 public class Ejemplo2Complemento {
4     public static void main(String[] args) {
5
6         int i = 15;
7         int j =20;
8
9         String mensaje = "";
10
11         if(i<j){
12             mensaje = "i es menor que j";
13         }
14
15         if(i>=j){
16             mensaje = "i es mayor o igual que j";
17         }
18
19         System.out.println(mensaje);
20     }
21 }
22
```

```
1 package ejs2condicionalesifelse;
2
3 public class Ejemplo2Complemento2 {
4     public static void main(String[] args) {
5
6         int i = 15;
7         int j =20;
8
9         String mensaje = "";
10
11         if(i<j){
12             mensaje = "i es menor que j";
13         }
14         else{
15             mensaje = "i es mayor o igual que j";
16         }
17
18         System.out.println(mensaje);
19     }
20 }
```

➤ Allí estamos representando el mismo flujo con dos variantes de if distintos.

1.2.3 VARIANTE 3: Sentencia IF - ELSE IF - ELSE

La última variante de la sentencia, es la de poder ir seleccionando según varias condiciones en caso de que alguna no se cumple. De esta manera, el enunciado sería de la siguiente manera “Si pasa X, entonces hacer J, sino, si pasa Y entonces hacer H, sino, si pasa Z hacer K, si no se cumple ninguna hacer M”.

La sintaxis, es la siguiente:

```
if (XXX){  
  
    } else if (YYY){  
  
    } else{  
    }
```

Donde:

- XXX es una condición que devuelve un valor booleano: true o false (para decidir si se cumple o no).
- YYY es otra condición que devuelve un valor booleano.
- Dentro de las primeras llaves, debemos colocar el código que queremos que se ejecuta si la condición XXX es verdadera, luego dentro de las segundas llaves, colocamos el código que queremos que se ejecute si: no se cumple XXX y se cumple YYY. Luego, si ninguna se cumple, tenemos código en el bloque else para ejecutar algo alternativo.
- El flujo de ejecución, para esta sentencia es el siguiente:
 - Se evalúa XXX, si es true, se ejecuta el código dentro de ese bloque. No se pregunta por ninguna otra condición y se sale del conjunto de if.
 - Si no se cumple XXX, se evalúa YYY. Si esta es true, se ejecuta el código del segundo bloque.
 - Por último, si no se YYY, se ejecuta el bloque de código que se encuentra dentro del else.

Veamos algunos ejemplos:

Ejemplo 1

```
Ejemplo1.java x
1 package ejs2condicionalesifelseifelse;
2
3 public class Ejemplo1 {
4     public static void main(String[] args) {
5
6         int i = 15;
7         int j = 20;
8
9         String mensaje = "";
10
11         if(i<j){
12             mensaje = "i es mejor que j";
13         }else if(i == j){
14             mensaje = "i es igual que j";
15         }else{
16             mensaje = "i es mayor que j";
17         }
18
19         System.out.println(mensaje);
20     }
21 }
22
23
```

Ejemplo 2

```
Ejemplo2.java x
1 package ejs2condicionalesifelseifelse;
2
3 public class Ejemplo2 {
4     public static void main(String[] args) {
5
6         int i = 15;
7
8         String mensaje = "";
9
10        if(i<5){
11            mensaje = "i es menor que 5";
12        }else if(i>=5 && i<15){
13            mensaje = "i está entre 5 y 14";
14        }else if(i>=15 && i<25){
15            mensaje = "i esta entre 15 y 24";
16        }else{
17            mensaje = "i es mayor o igual que 25";
18        }
19
20        System.out.println(mensaje);
21
22    }
23 }
24
```

➤ Como se puede ver en el ejemplo, podemos agregar cuantos "else if"

necesitemos. En el ejemplo, podemos ver que hay únicamente 2, pero podríamos tener más de ser necesario.

Ejemplo 3

```
Ejemplo3.java x
1 package ej2condicionalesifelseifelse;
2
3 public class Ejemplo3 {
4     public static void main(String[] args) {
5
6         int i = 200;
7
8         String mensaje = "por defecto";
9
10        if(i<5){
11            mensaje = "i es menor que 5";
12        }else if(i>=5 && i<15){
13            mensaje = "i está entre 5 y 14";
14        }else if (i>=15 && i<25){
15            mensaje = "i está entre 15 y 24";
16        }
17
18        System.out.println(mensaje);
19
20    }
21 }
22
```

- Este tercer ejemplo, nos ilustra que podemos prescindir de la sentencia “else”.
- ¿Que se imprime en este ejemplo en la sentencia print del final? La respuesta es “por defecto”. Esto se debe a que cuando se evalúan todas las condiciones, ninguna nos da verdadero, por ende, la variable no se le modifica el valor.

Importante:

- La anidación de sentencias if, else if y else siempre comienza con el if.
- Una vez que tenemos el primer if, podemos implementar cuantos else if consideremos necesarios.
- Opcionalmente, podemos implementar UN bloque else, que se ejecuta cuando no se cumple ninguna condición. ¿Porque solo implementamos un bloque else? Tener más de uno no tiene sentido si pensamos en el flujo del programa.

1.3 Operaciones con variables String

Java nos provee de funciones ya implementadas para trabajar con variables de tipo de dato String. Una de estas funciones es el equals: Se utiliza `varStr1.equals(varStr2)` y esto devuelve true, si `varStr1` es igual a `varStr2` o false en caso contrario.

Ejemplo 1

```
Ejemplo1.java x
1 package ej2operacionesconstring;
2
3 public class Ejemplo1 {
4     public static void main(String[] args) {
5
6         String var1 = "hola";
7         String var2 = "hola";
8         String var3 = "chau";
9
10        if(var1.equals(var2)){
11            System.out.println("var1 y var2 son iguales");
12        }
13
14        if(var1.equals(var3)){
15            System.out.println("var1 y var3 son iguales");
16        }else{
17            System.out.println("var1 y var3 son distintas");
18        }
19    }
20 }
21
22
```

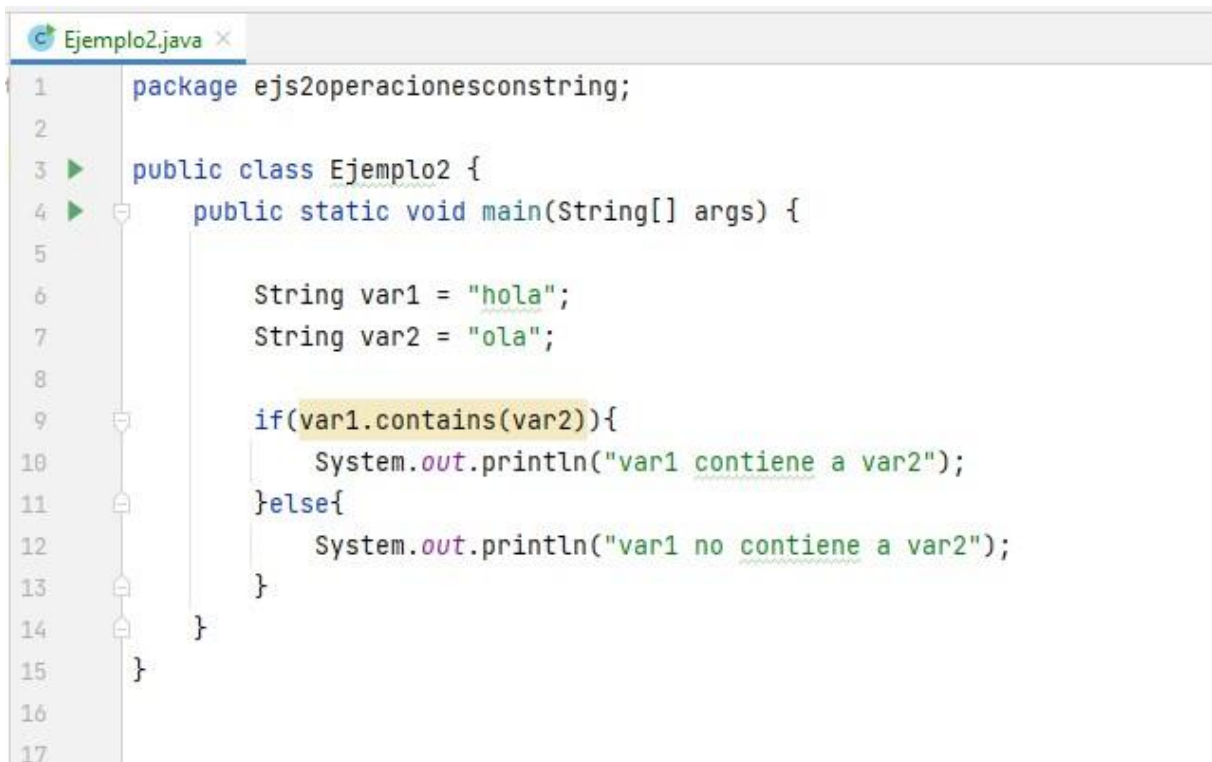
Al ejecutar este programa se imprime:

var 1 y var 2 son iguales var
1 y var 3 son distintas

Luego tenemos, la función contains:

Se utiliza `varStr1.contains(varStr2)` y esto devuelve true, si el valor de `varStr1` contiene el valor de `varStr2` o false en caso contrario.

Ejemplo 2



```
1 package ejs2operacionesconstring;
2
3 public class Ejemplo2 {
4     public static void main(String[] args) {
5
6         String var1 = "hola";
7         String var2 = "ola";
8
9         if(var1.contains(var2)){
10             System.out.println("var1 contiene a var2");
11         }else{
12             System.out.println("var1 no contiene a var2");
13         }
14     }
15 }
16
17
```

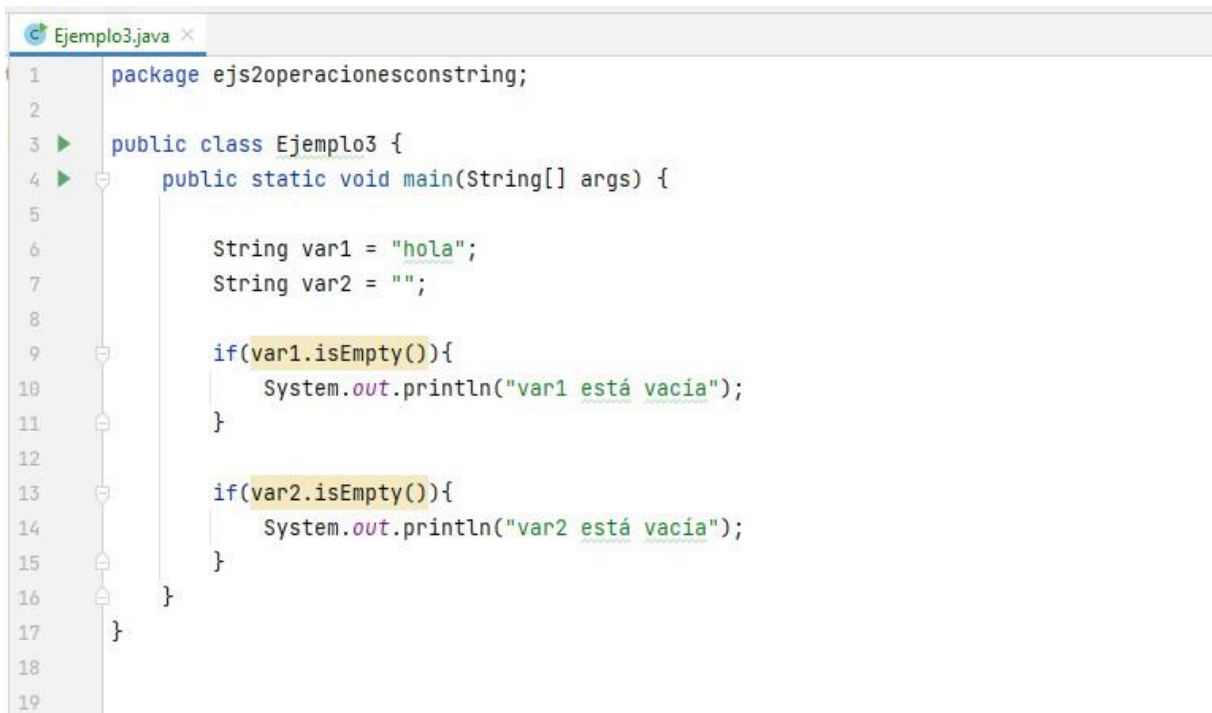
Al ejecutar este programa se imprime:

var 1 contiene a var 2

Otra función de Java, utilizada en el trabajo con Strings es isEmpty:

Se utiliza `varStr1.isEmpty()` y esto devuelve `true`, si el valor de `varStr1` es vacío o `false` en caso contrario.

Ejemplo 3



```
1 package ejs2operacionesconstring;
2
3 public class Ejemplo3 {
4     public static void main(String[] args) {
5
6         String var1 = "hola";
7         String var2 = "";
8
9         if(var1.isEmpty()){
10             System.out.println("var1 está vacía");
11         }
12
13         if(var2.isEmpty()){
14             System.out.println("var2 está vacía");
15         }
16     }
17 }
18
19
```

Al ejecutar este programa se imprime:

var 2 esta vacia

Operador Java Condicional Ternario

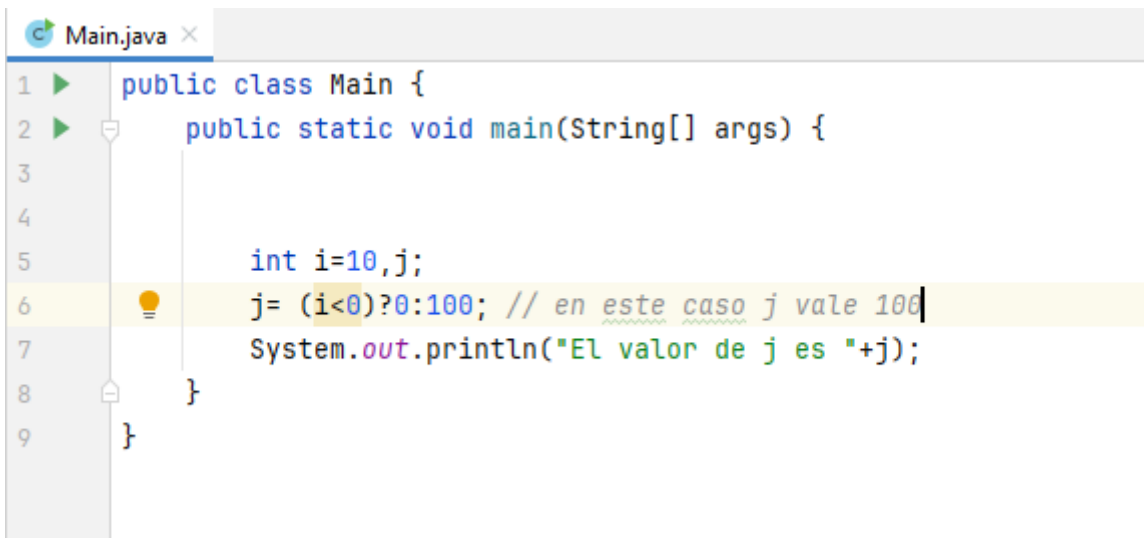
Java nos provee el operador ternario para sustituir las sentencias if-else, a veces el código puede quedar un poco menos claro, pero de esta forma podemos ahorrar líneas de código.

Sintaxis:

variable = (condición) ? valor1 : valor2 ;

Por ejemplo, en este caso si la condición es verdadera en variable se va a guardar el valor1, sino lo es se guarda el valor2.

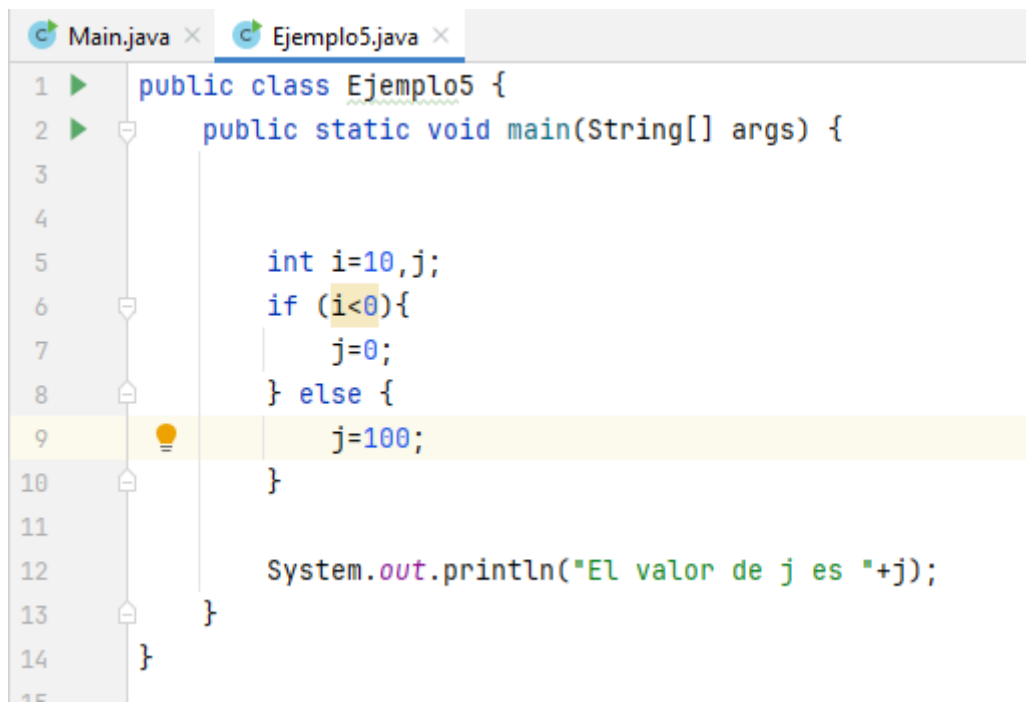
Ejemplo 4



```
1 public class Main {  
2     public static void main(String[] args) {  
3  
4  
5         int i=10,j;  
6         j= (i<0)?0:100; // en este caso j vale 100  
7         System.out.println("El valor de j es "+j);  
8     }  
9 }
```

Este mismo ejemplo realizado con las sentencias if-else sería de la siguiente forma:

Ejemplo 5



```
1 public class Ejemplo5 {
2     public static void main(String[] args) {
3
4
5         int i=10,j;
6         if (i<0){
7             j=0;
8         } else {
9             j=100;
10        }
11
12        System.out.println("El valor de j es "+j);
13    }
14 }
```

Podemos ver otros ejemplos, incluso podemos poner otro operador ternario dentro de las opciones a tomar.

Ejemplo 6

```
Ejemplo6.java x
1 public class Ejemplo6 {
2     public static void main(String[] args) {
3         int a=1, b=2, c=3;
4         c+= (a>0 && a<=10)? ++a:a/b;
5         System.out.println("El valor del primer c es "+c);
6
7         // otro caso
8         a=50;
9         b=10;
10        c=20;
11
12        c+= (a>0 && a<=10)? ++a:a/b;
13        System.out.println("El valor del segundo c es "+c);
14
15        // tiene un operador ternario dentro de otro operador ternario
16        c = (a>60)?(a<30)?a-c:b:a;
17        System.out.println("El valor del tercer c es "+c);
18    }
19 }
```

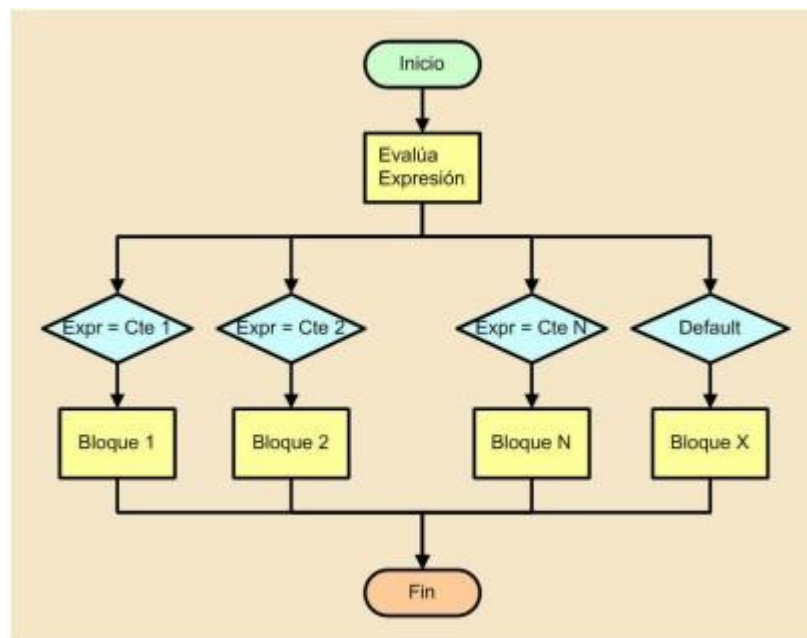
```
El valor del primer c es 5
El valor del segundo c es 25
El valor del tercer c es 50
```

1.4 Sintaxis de SWITCH

En esta sección, veremos la sentencia condicional llamada Switch. La misma, nos sirve para: dado el valor de una variable (solo los tipos de datos: byte,short,char,int o String), que porción de código (bloque) queremos ejecutar.

La sintaxis de la instrucción es la siguiente:

```
switch (variable){  
    case (XXX): break;  
    case (YYY): break;  
    default: break;  
}
```



Donde:

- “variable” es una variable que debe tener tipo byte,short,char,int o String.
- XXX es un valor fijo del mismo tipo que la “variable”.
- YYY es un valor distinto de XXX, pero también del mismo tipo que “variable”.

Veamos algunos ejemplos:

Ejemplo 1

```
Ejemplo1.java x
1  package ejs2switch;
2
3  public class Ejemplo1 {
4  public static void main(String[] args) {
5
6      int var = 10;
7
8      double resultado = 0;
9
10     switch (var){
11
12         case 5:
13             int otraVar = 56;
14             resultado = 34.22 + otraVar;
15             break;
16
17         case 15:
18             resultado = 34.22;
19             break;
20
21         case 10:
22             resultado = 34.22 + 56;
23             System.out.println(resultado);
24             break;
25
26         default:
27             System.out.println("No tiene ni valor 5, ni 15, ni 10");
28     }
29 }
30 }
31
```

➤ Opcionalmente podemos utilizar el default. Donde, el bloque “default”, se ejecutará en caso de no coincidir con ninguno de los valores anteriores.

➤ El flujo de ejecución del código anterior, sería algo así: primero evalúo si la variable “variable” tiene el valor 5. De tenerlo, pasaremos a ejecutar las tres sentencias que se encuentran bajo este valor. De lo contrario se continúa inspeccionando los siguientes “case”. De no encontrar un case con el valor que tiene la variable, se ejecuta todo lo que se encuentre bajo default.

➤ Todo conjunto de sentencias dentro de un case, debe terminar con la palabra break y un punto y coma.