



# Programación 1

Selectivas e iterativas en JAVA

# Sentencia if

¿Qué es una expresión lógica?

```
if (expresionLogica){  
    Sentencia_1;  
    Sentencia_2;  
    ...  
    ...  
    Sentencia_N;  
}
```

```
SI (dia== domingo) ENTONCES{  
    Miro una serie;  
    Practico programación;  
}
```



```
SI ((dia== jueves) && (dia != feriado)) ENTONCES{  
    Prendo la compu;  
    Me conecto a la meet;  
    Me pongo a escuchar al profe;  
}
```



# Sentencia if - ejemplo



```
int a = 2;
int c = 3;
if (a < c){
    System.out.println(a+" es menor que " + c);
    a = 0;
}
System.out.println("a vale: " + a);
```

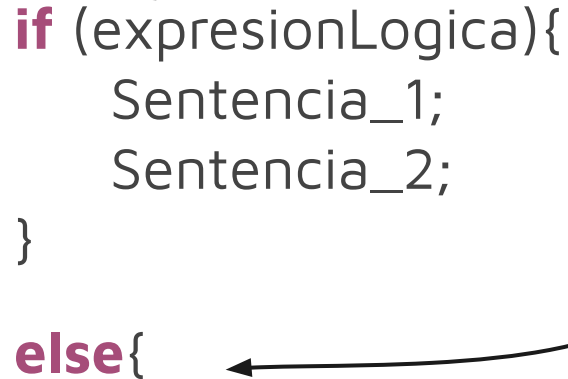
¿Qué sale por la consola?

```
int a = 2;
int c = 3;
a=a+1;
if (a < c){
    System.out.println(a+" es menor que " + c);
    a = 0;
}
System.out.println("a vale: " + a);
```

¿Qué sale por la consola?

# Sentencia if-else

Da la idea de exclusividad o pasa una cosa o la otra, nunca las dos a la vez



```
if (expresionLogica){  
    Sentencia_1;  
    Sentencia_2;  
}  
  
else{  
    Sentencia_3;  
    Sentencia_4;  
}
```

```
SI (es semana virtual) ENTONCES {  
    Me quedo en casa;  
    Me conecto a la meet;  
}  
SINO {  
    Voy al campus;  
}
```

# Sentencia if-else - ejemplo

```
int a = 6;
int c = 3;
if (a < c){
    System.out.println(a + " es menor que " + c);
    a = 0;
}
else{
    System.out.println(a + " no es menor que " + c);
}
```

¿Puedo asegurar que c es menor que a?

En memoria RAM

inicial	final
a=6	a=?
c=3	c=?

# Sentencia if-else if



```
if (expresionLogica){  
    Sentencia_1;  
    Sentencia_2;  
}
```

```
else if (expresionLogica2){  
    Sentencia_3;  
    Sentencia_4;  
}
```

También da la idea de exclusividad o pasa una cosa o la otra, nunca las dos a la vez pero abre una nueva rama de posibilidades

```
SI (es semana virtual) ENTONCES {  
    Me quedo en casa;  
    Me conecto a la meet;  
}  
SINO SI ((dia==martes) || (dia==jueves)) ENTONCES {  
    Preparo mate;  
    Curso programación;  
}
```

¿Qué pasa si no es la semana virtual y no es martes ni jueves?

# Sentencia if-else -if - ejemplo

```
int a = 3;
int c = 3;
if (a < c){
    System.out.println(a + " es menor que " + c);
    a = 0;
}
else if( a == c){
    System.out.println(a + " es igual a " + c);
}
```

En memoria RAM	
inicial	final
a=3	a=?
c=3	c=?

# Sentencia if-else if-else

```
if (expresionLogica){  
    Sentencia_1;  
    Sentencia_2;  
}  
  
else if (expresionLogica2){  
    Sentencia_3;  
    Sentencia_4;  
}  
  
else Sentencia_5;
```



También da la idea de exclusividad o pasa una cosa o la otra, nunca las N a la vez y se puede seguir anidando

```
SI (es semana virtual) ENTONCES {  
    Me quedo en casa;  
    Me conecto a la meet;  
}  
  
SINO ((dia==martes) || (dia== jueves)) ENTONCES {  
    Preparo Mate;  
    Curso programación;  
}  
  
SINO {  
    Voy a correr;  
}
```

¿Entonces, cuándo voy a correr?



# Sentencia if-else if - else - ejemplo

```
int a = 2;
int c = 3;
if (a < c){
    System.out.println(a + " es menor que " + c);
    c=45;
}
else if( a == c){
    System.out.println(a + " es igual a " + c);
}
else
    System.out.println(a + " es mayor que " + c);
```

En memoria RAM

inicial	final
a=2	a=?
c=3	c=?

# Sentencia if-else if - else - ejemplo

En memoria RAM

inicial	final
a=2	a=?
c=3	c=?

```
int a = 2;
```

```
int c = 3;
```

```
if (a < c){
```

```
    System.out.println(a + " es menor que " + c);
```

```
    a = 0; c = 0;
```

```
}
```

```
else if( a == c){
```

```
    System.out.println(a + " es igual a " + c);
```

```
}
```

```
else System.out.println(a + " es mayor que " + c);
```

¿Qué pasa aquí?

Entonces...

¿Entra luego aquí?

# Sentencia if-else if - else - ejemplo

```
int a = 2;
int c = 3;
if (a < c){
    System.out.println(a + " es menor que " + c);
    a = 0; c = 0;
}
else if( a == c){
    System.out.println(a + " es igual a " + c);
    a = 0; c = 0;
}
else { System.out.println(a + " es mayor que " + c);
    a = 0; c = 0; }
```

En memoria RAM	
inicial	final
a=2	a=?
c=3	c=?

¿Qué pasa aquí?

# Sentencia if-else if - else - ejemplo

```
int a = 2;
int c = 3;
if (a < c){
    System.out.println(a + " es menor que " + c);
}
else if( a == c){
    System.out.println(a + " es igual a " + c);
}
else { System.out.println(a + " es mayor que " + c); }
```

a = 0; c = 0;

En una línea luego de la estructura if-else

En memoria RAM	
inicial	final
a=2	a=?
c=3	c=?

# Sentencia if-else if - else - ejemplo

```
int a = 2;
int c = 3;
if (a < c){
    System.out.println("Resultado de la comparación:");
    System.out.println(a + " es menor que " + c);
}
else if( a == c){
    System.out.println("Resultado de la comparación:");
    System.out.println(a + " es igual a " + c);
}
else {
    System.out.println("Resultado de la comparación:");
    System.out.println(a + " es mayor que " + c); }
```

En memoria RAM	
inicial	final
a=2	a=?
c=3	c=?

¿Qué pasa aquí?

# Sentencia if-else if - else - ejemplo

```
int a = 2;
int c = 3;
System.out.println("Resultado de la comparación:");
if (a < c){
    System.out.println(a + " es menor que " + c);
}
else if( a == c){
    System.out.println(a + " es igual a " + c);
}
else {
    System.out.println(a + " es mayor que " + c);
}
```

En una línea antes de la estructura if-else

En memoria RAM	
inicial	final
a=2	a=?
c=3	c=?

//Dado un número entero ingresado por el usuario: imprimir A si el numero es multiplo de 2, o imprimir  
//B si el número es múltiplo de 7, o imprimir C si el número es múltiplo de 2 y de 3, o imprimir el número

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Clase_2_Ejemplo_1 {
    public static void main (String [] args) {
        int numero = 0; //DECLARACION DE CONSTANTES Y VARIABLES
        try { //CARGA DE DATOS DE ENTRADA
            BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
            System.out.println ("Ingrese número: ");
            numero = Integer.valueOf(entrada.readLine());
            if ((numero%2)==0) {    //PROCESAMIENTO Y SALIDA
                System.out.println("A");
            }
            else if ((numero%7)==0) {    //PROCESAMIENTO Y SALIDA
                System.out.println("B");
            }
            else if (((numero%2)==0)&&((numero%3)==0)) {
                System.out.println("C");
            }
            else {
                System.out.println("El número es:" + numero);
            }
        }
        catch (Exception exc ) {
            System.out.println("Hubo algún error:" + exc );
        }
    }
}
```

```
//Dado un caracter ingresado por el usuario: imprimir si es caracter minúscula, o imprimir si es
//carácter mayúscula, o imprimir no es caracter letra
```

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Clase_2_Ejemplo_2 {
    public static void main(String[] args) {
        char caracter;
        try{
            BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
            System.out.println("Ingrese un caracter :");
            caracter = entrada.readLine().charAt(0);
            if (('a'<=caracter)&&(caracter<='z')) {
                System.out.println("Es carácter minúscula");
            }
            else if (('A'<=caracter)&&(caracter<='Z')) {
                System.out.println("Es carácter mayúscula");
            }
            else {
                System.out.println("No es carácter letra");
            }
        }
        catch (Exception exc){
            System.out.println(exc);
        }
    }
}
```



# Sentencia switch



```
switch (variable){  
    case valor_1: Sentencia_1; break;  
    case valor_2: Sentencia_2; break;  
    case valor_3: {  
        Sentencia_3;  
        Sentencia_4;  
        break;  
    };  
    default: Sentencia_5;  
}
```

# Sentencia switch - ejemplo



```
char letra = 'c';  
switch (letra){  
    case 'a': System.out.println("Es una a"); break;  
    case 'e': System.out.println("Es una e"); break;  
    case 'i': System.out.println("Es una i"); break;  
    case 'o': System.out.println("Es una o"); break;  
    case 'u': System.out.println("Es una u"); break;  
    default: System.out.println("No es una vocal");  
}
```

# Sentencia switch



```
switch (variable){  
    case valor_1: case valor_2: Sentencia_1;break;  
    case valor_3: {  
        Sentencia_3;  
        Sentencia_4;  
        break;  
    };  
    default: Sentencia_5;  
}
```

¿Podemos modificar el ejemplo anterior e informar cuando una letra es vocal o consonante?

¿Cómo sería?

# Sentencia switch - ejemplo



```
char letra = 'c';  
switch (letra){  
    case 'a': case 'e': case 'i': case 'o': case 'u': System.out.println("Es  
una vocal"); break;  
    default: System.out.println("Es una consonante");  
}
```

## Sentencia switch - ejemplo 2



```
int numero = 4;
switch (numero){
    case 1, 3, 5, 7, 9: System.out.println("Es un número impar menor
a 10"); break;
    case 0, 2, 4, 6, 8: System.out.println("Es un número par menor
a 10"); break
    default: System.out.println("Es un número mayor a 10 o menor a
0");
}
```



```
if (es_necesario())
```

**CORTE DE 15' Y SEGUIMOS  
CON ITERATIVAS**

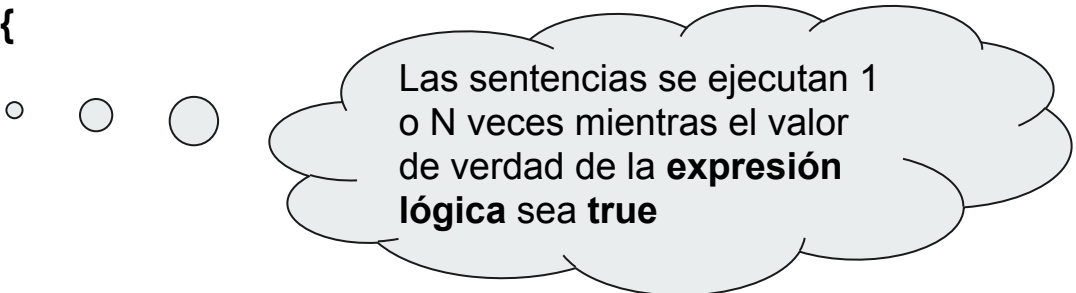
# Sentencias iterativas

Un bucle o sentencia repetitiva se utiliza cuando se requiere hacer una o un conjunto de tareas varias veces

En una sentencia repetitiva hay una **expresión lógica** (condición simple o múltiple) que:

- si es cierta, ejecuta las sentencias entre llaves, y posteriormente vuelve a verificar la expresión lógica de terminación.
- si es falsa, sale del bucle.

```
Ciclo (expresión lógica) {  
    sentencia 1;  
    sentencia n;  
    ...  
}
```



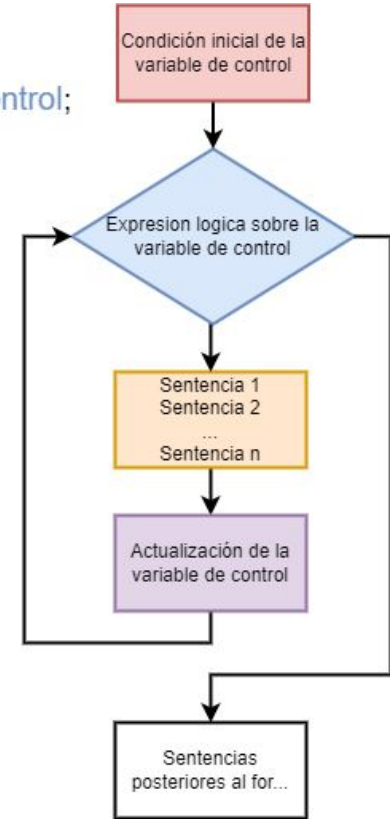
Las sentencias se ejecutan 1 o N veces mientras el valor de verdad de la **expresión lógica** sea **true**

The diagram illustrates the loop logic. It shows a code block for a loop. To the right of the code block, there are three small circles of increasing size, leading to a large thought bubble. Inside the thought bubble, it explains that the sentences are executed 1 or N times as long as the truth value of the logical expression is true.

# Sentencia for

```
for( condición inicial de variable de control; expresión lógica sobre variable de control;  
    actualización de la variable de control) {  
    sentencia_1;  
    sentencia_2;  
    ...  
    sentencia_n;  
}
```

La sentencia **for** SOLO se utiliza cuando se conoce exactamente la cantidad de iteraciones.





# Ejemplo



```
//Sentencia for
//Dado un número entero con valor inicial 5, imprimir la tabla de multiplicar de dicho número

public class Clase_2_Ejemplo_3 {
    public static void main(String args[]) {
        final int MAX = 10;
        final int MULTIPLO = 5;
        System.out.println("Tabla de multiplicar del" + MULTIPLO);
        // Se puede declarar o no la variable multiplicador dentro del inicio del for
        for (int multiplicador = 1; multiplicador <= MAX; multiplicador++) {
            System.out.println(MULTIPLO+" * "+multiplicador+" = "+(MULTIPLO * multiplicador));
        }
    }
}
```

# Ejemplo

//Sentencia for anidada

```
public class Clase_2_Ejemplo_4 {  
    public static void main(String args[]) {  
        final int MAX = 10;  
        final int MULTIPLO = 3;  
        System.out.println("Tablas de multiplicar del 1, 2 y 3");  
        for (int i = 1; i <= MULTIPLO; i++) {  
            System.out.println("Tabla de multiplicar del " + i);  
            for (int j = 1; j <= MAX; j++) {  
                System.out.println(j + " * " + i + " = " + j * i);  
            }  
        }  
    }  
}
```

# Ejemplo



Dentro de los bloques {...} de las sentencias iterativas hay sentencias que pueden ser o no otras sentencias iterativas.

```
public class Clase_2_Ejemplo_5 {  
    public static void main(String[] args) {  
        ...  
        while (...){  
            ...  
            ...  
            for( ; ; ){  
                ...  
                while (...){}  
            }  
        }  
        ...  
    }  
}
```

```
//Hacer un programa que mientras el usuario cargue un número entero != 0 imprima
//la tabla de multiplicar de dicho número
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Clase_2_Ejemplo_5 {
    public static void main(String[] args) {
        int numero = 9;
        final int MAX=10;
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        while (numero != 0) {
            try {
                System.out.println("Ingrese un numero entero (0 para salir): ");
                numero = Integer.valueOf(entrada.readLine());
                if (numero != 0) {
                    for (int multiplicador=1; multiplicador <= MAX; multiplicador++)
                        System.out.println(numero+"*"+multiplicador+"="+ (numero * multiplicador));
                }
            }
            catch (Exception exc) {
                System.out.println(exc);
            }
        }
    }
}
```

```

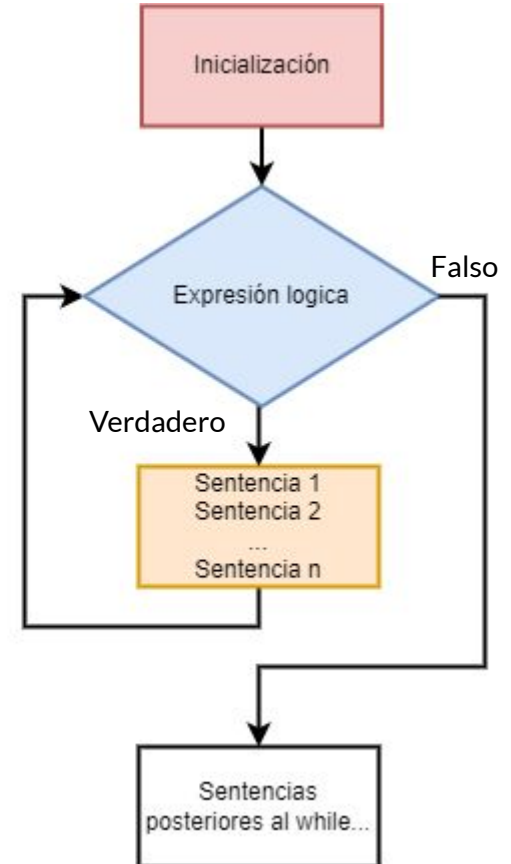
/*Hacer un programa que mientras que el usuario cargue un número entero entre 0 y 1000 (no incluidos),
guarde el menor de los números ingresados. Al finalizar el ciclo imprima el menor por pantalla*/
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Clase_2_Ejemplo_6 {
    public static void main(String[] args) {
        final int MIN = 0, MAX = 1000;
        int numero = 0, menor = MAX;
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        try{
            System.out.println("Ingrese un número entre "+MIN+" y "+MAX+", con otro valor sale del ciclo");
            numero = Integer.valueOf(entrada.readLine());
            while ((MIN < numero)&&(numero < MAX)){
                if (numero<menor){
                    menor=numero;
                }
                System.out.println("Ingrese un número entre "+MIN+" y "+MAX+", con otro valor sale del ciclo");
                numero = Integer.valueOf(entrada.readLine());
            }
        }
        catch (Exception exc){
            System.out.println(exc);
        }
        if (menor != MAX){
            System.out.println("El menor ingresado es: " + menor);
        }
    }
}

```

# Sentencia While

```
[inicialización]
while( expresión lógica ) {
    sentencia_1;
    sentencia_2;
    ...
    sentencia_n;
}
```

La sentencia while se utiliza cuando se conoce exactamente o **NO** la cantidad de iteraciones.



# Sentencia While



- Una sentencia iterativa se utiliza cuando se requiere realizar una o muchas tareas varias veces (varias veces puede ser ninguna, una o más).
- En la sentencia iterativa **while (expresión lógica){...}** hay una expresión lógica que:
  - si es **verdadera**, **ejecuta** las sentencias entre llaves, y luego vuelve al **inicio** de la sentencia iterativa para evaluar nuevamente la expresión lógica.
  - si es **falsa**, **sale** de la iteración.
- Mientras la **expresión lógica** sea verdadera va a ejecutarse lo que está dentro de las {...} en el orden en que aparecen.
- Algún valor de las **variables** de la expresión lógica deberá **cambiar** en algún ciclo de la repetición dentro de las sentencias en las {...}, sino se producirá un **ciclo infinito**.

# Ejemplo While



*/\*Dado un número entero con valor inicial 1, hacer una iteración que haga incrementar el número de a uno hasta un valor MAX = 4 (constante). Mientras itera deberá imprimir número \*/*

```
public class Clase_2_Ejemplo_7 {  
    public static void main (String [] args) {  
        final int MAX = 4;  
        int numero = 1;  
        while (numero <= MAX){  
            System.out.println("El numero es: " + numero);  
            //al cambiar de valor el número significa que el valor de la  
            //expresión lógica va a cambiar  
            numero++;  
        }  
    }  
}
```



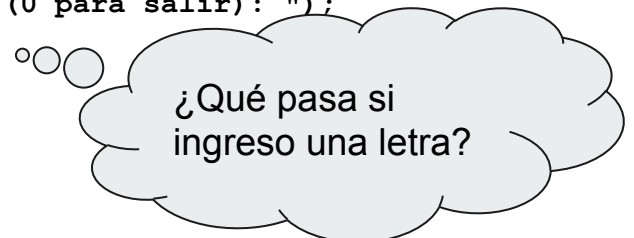
# A tener en cuenta

La utilización de sentencias iterativas implica **reubicar** sentencias declarativas como la de **BufferedReader**, para no iterar con una declaración dentro de las llaves {...} del ciclo y así evitar que en cada **ciclo** se **declare** la misma variable.

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Clase_2_Ejemplo_8 {
    public static void main(String[] args) {
        //ubicar el buffer entrada cerca de la región de declaración de variables
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        try{
            while (...){
            }
        }
        catch (Exception exc){
            System.out.println(exc);
        }
    }
}
```

# Ejemplo try-while

```
//Hacer un programa que mientras el usuario cargue un número entero distinto de 0 lo imprima
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Clase_2_Ejemplo_9{
    public static void main(String[] args) {
        int numero = 0;
        //la declaración del buffer entrada la ubico al principio junto con las otras declaraciones
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        try{
            //el usuario carga un valor la primera vez
            System.out.println("Ingrese un numero entero (0 para salir): ");
            numero = Integer.valueOf(entrada.readLine());
            while (numero != 0){
                //si número es distinto de 0 lo imprime, vuelve a pedir su carga, y regresa al while
                System.out.println("El valor es: " + numero);
                System.out.println("Ingrese un numero entero (0 para salir): ");
                numero = Integer.valueOf(entrada.readLine());
            }
        }
        catch (Exception exc){
            System.out.println(exc);
        }
    }
}
```

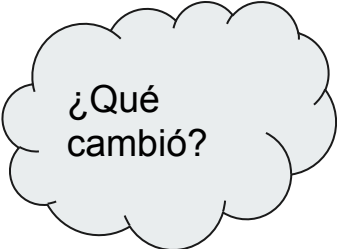


¿Qué pasa si ingreso una letra?

# Ejemplo while-try

```
//Hacer un programa que mientras el usuario cargue un número entero distinto de 0 lo imprima
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Clase_2_Ejemplo_10{
    public static void main(String[] args) {
        int numero = 9;
        //la declaración del buffer entrada la ubico al principio junto con las otras declaraciones
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        while (numero != 0) {
            try {
                System.out.println("Ingrese un numero entero (0 para salir): ");
                numero = Integer.valueOf(entrada.readLine());
                if (numero != 0)
                    System.out.println("El valor es: " + numero);
            }
            catch (Exception exc) {
                System.out.println(exc);
            }
        }
    }
}
```

¿Cómo podemos hacer si queremos  
lo imprima solo si es par?



¿Qué  
cambió?

## Ejemplo capital acumulado



Para este problema deberá utilizar todos los tipos de sentencias desarrolladas hasta el momento.

Utilizando una sentencia repetitiva calcular el capital acumulado a 10 años para un capital inicial de \$100 y una tasa de interés de 4% anual.

# Ejemplo capital acumulado

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Clase_2 Ejemplo_11 {
    public static void main(String args[]) {
        final int MAX = 10;
        final int interes = 4;
        double capital = 100.0;
        int anios = 1;
        while (anios <= MAX) {
            capital += capital*interes/100;
            anios++;
        }
        System.out.println("Capital final es = " + capital);
    }
}
```

## Ejemplo valor positivo válido



Para este problema deberá utilizar todos los tipos de sentencias desarrolladas hasta el momento.

Ingresar un valor positivo **válido**.

# Ejemplo valor positivo válido

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Clase_2_Ejemplo_12 {
    public static void main(String args[]) {
        int valor = 0;
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        while (valor <= 0) { // los valores positivos no incluyen al 0
            try { // try define un bloque de manejo de posibles excepciones
                System.out.println("Ingrese valor positivo: ");
                valor = Integer.valueOf(entrada.readLine());
                System.out.println("El valor ingresado es: " + valor);
            } catch (Exception exc) {
                System.out.println(exc);
            }
        }
    }
}
```

## Ejemplo valor válido



Para este problema deberá utilizar todos los tipos de sentencias desarrolladas hasta el momento.

Ingresar un valor entero **válido**.



# Ejemplo valor válido

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Clase_2_Ejemplo_13 {
    public static void main(String args[]){
        int valor = 0;
        boolean esValido = false;
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));
        while (!esValido) { // mientras no haya ingresado un nro válido
            try { // try define un bloque de manejo de posibles excepciones
                System.out.println("Ingrese un valor entero: ");
                valor = Integer.valueOf(entrada.readLine());
                System.out.println("El valor ingresado es: " + valor);
                esValido = true;
            } catch (Exception exc) {
                System.out.println("El valor ingresado no es válido");
            }
        }
    }
}
```

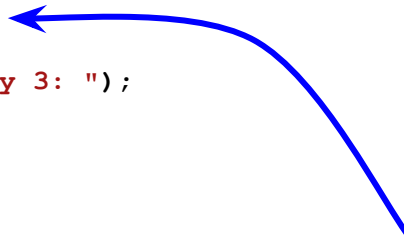
# Ejemplo de problema



Para este problema deberá utilizar todos los tipos de sentencias desarrolladas hasta el momento.

Hacer un programa que dado un valor ingresado por el usuario entre 1 y 3 inclusive (**si ingresa otro valor termina**), imprima como salida “Bajo” en el caso de que ingrese 1, “Medio” si ingresa 2, y “Alto” si ingresa 3.

```
public class Clase_2_Ejemplo_14 {  
    public static void main(String[] args) {  
        final int MINIMO = 1;  
        final int MAXIMO = 3;  
        int valor = 0;  
        BufferedReader entrada = new BufferedReader(new InputStreamReader(System.in));  
        while (!(valor >= MINIMO) && (valor <= MAXIMO)) {  
            try {  
                System.out.println("Ingrese integer entre 1 y 3: ");  
                valor = Integer.valueOf(entrada.readLine());  
                switch (valor) {  
                    case 1:  
                        System.out.println("Bajo");  
                        break;  
                    case 2:  
                        System.out.println("Medio");  
                        Break;  
                    case 3:  
                        System.out.println("Alto");  
                        break;  
                }  
            }  
            catch (Exception e) {  
                System.out.println(e);  
            }  
        }  
    }  
}
```



Se puede escribir de otra forma  
esta expresión lógica?

# Algunos tips



- Cuando tengo que tomar una decisión con dos resultados solo uso un if-else
- Cuando hay más posibilidades se puede anidar más
- Si hay sentencias que se repiten en la rama verdadera y falsa, se deben sacar afuera de la estructura.
- En cada caso del switch, siempre poner break sino sigue evaluando el siguiente caso.
- Si quiero obligar al usuario que ingrese un valor o rango, usar while y el try dentro, por si hay error
- La sentencia while (expresión lógica) {...} se utiliza cuando se conoce exactamente o no la cantidad de iteraciones.
- La sentencia for ( ; ; ) {...} se utiliza cuando se conoce exactamente la cantidad de iteraciones.
- No modificar la variable de control del for para forzar el corte. No forzar el corte con un break ni con return sea en for o while (aprender a armar bien la expresión lógica para que corte solo).
- Por ejemplo cuando la expresión lógica tiene una variable que se carga dentro de las {...} o su valor es resultado de un cálculo (no de un incremento o decremento) usar while (expresión lógica) {...} ya que se desconoce cuántas iteraciones hará la sentencia.