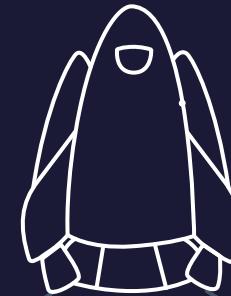


Programa académico CAMPUS



MODULO JAVA Sesión 2: Operadores y estructuras de decisión

Trainer Carlos H. Rueda C.

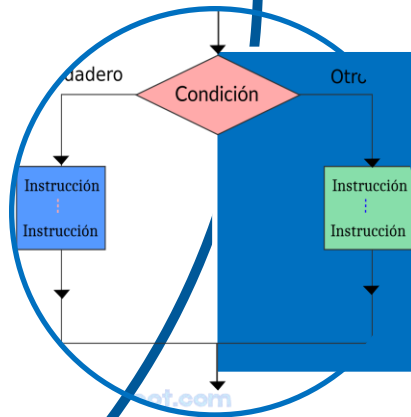




TEMARIO



Operadores



Estructuras de
decisión



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

Los operadores se utilizan para realizar operaciones (valga la redundancia) en variables y valores. Son representados con símbolos especiales y se clasifican en los siguientes grupos:



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

1. Operadores de asignación
2. Operadores aritméticos
3. Operadores de asignación 2.0
4. Operadores Unarios
5. Operadores de igualdad y relacionales
6. Operadores lógicos



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

1. Operadores de asignación

El operador Java más sencillo es el operador de asignación. Mediante este operador se asigna un valor a una variable. El operador de asignación es el símbolo igual. Este operador es independiente del tipo de variable que se opera porque su función se limita como su nombre lo dice a la asignación.

Ejemplo:

```
int valor = 123;  
boolean flag = true;  
String cadena;  
cadena = "Hello";
```



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

2. Operadores aritméticos

La siguiente tabla resume los diferentes tipos de operadores aritméticos

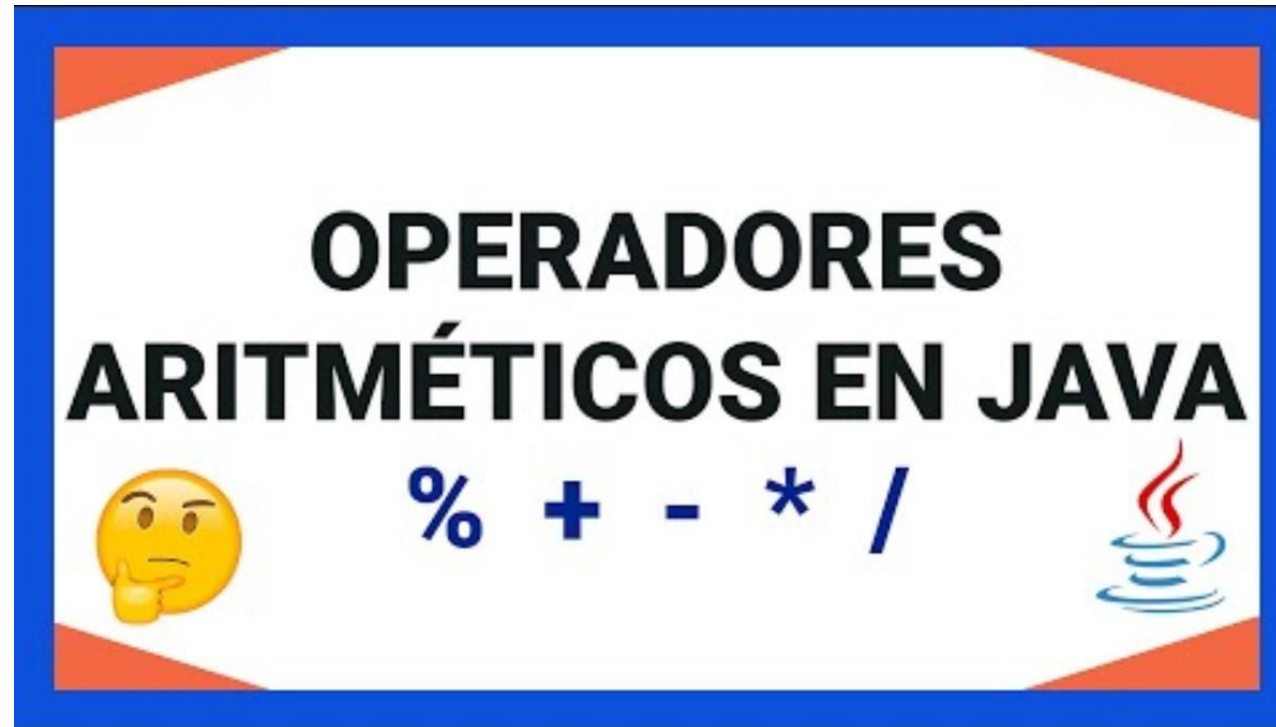
Operador	Descripción	Ejemplo
+ (Adición)	Agrega valores a ambos lados del operador.	$A + B$ dará 30
- (Resta)	Resta el operando de la derecha del operando de la izquierda.	$A - B$ dará -10
* (Multiplicación)	Multiplica valores a ambos lados del operador.	$A * B$ dará 200
/ (División)	Divide el operando de la izquierda por el operando de la derecha.	B/A dará 2
% (Módulo)	Divide el operando de la izquierda por el operando de la derecha y devuelve el resto.	$B \% A$ dará 0
++ (incremento)	Aumenta el valor del operando en 1.	$B++$ da 21
-- (Disminución)	Disminuye el valor del operando en 1.	$B--$ da 19



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

1. Operadores aritméticos

El siguiente video explica en Java como se implementan los operadores aritméticos.



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

1. Operadores de asignación 2.0

Aparte del operador de asignación de igualdad ($=$), existen otros operadores que me permite operar y asignar. La siguiente tabla los resumen

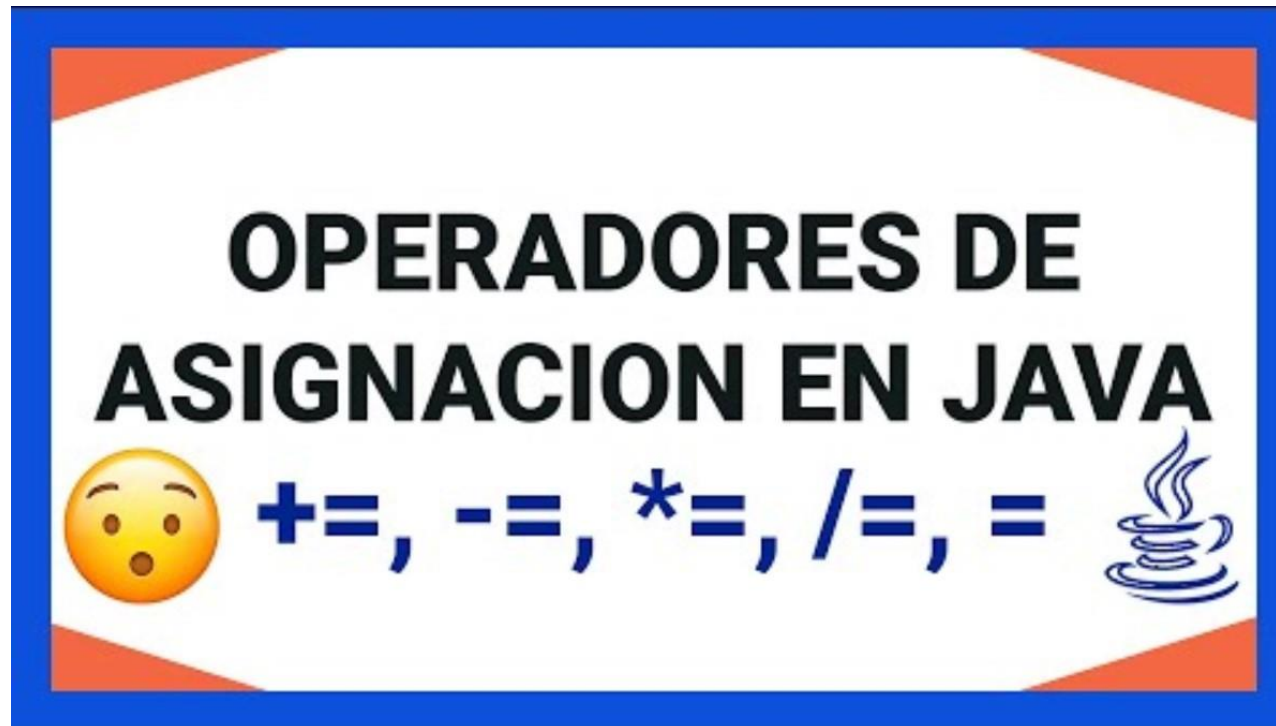
Operador	Nombre	Ejemplo	Equivalencia	Descripción
$=$	Asignación	$a = b$	$a = b$	Se utiliza para asignar un valor del operando del lado derecho al izquierdo.
$+=$	Suma combinada	$a += b$	$a = a + b$	Este operador agrega el operando del lado derecho a la variable izquierda y asigna el resultado al operando izquierdo.
$-=$	Resta combinada	$a -= b$	$a = a - b$	Este operador se utiliza para restar el valor del lado derecho de la variable y asignar el resultado al operando izquierdo.
$*=$	Producto combinado	$a *= b$	$a = a * b$	Este operador se utiliza para multiplicar el operando izquierdo por el derecho y asignar el resultado a la variable del lado izquierdo.
$/=$	División combinada	$a /= b$	$a = a / b$	Se utiliza para dividir el operando del lado izquierdo por el de la derecha y asignar el resultado a la variable de la izquierda.
$\% =$	Resto combinado	$a \% = b$	$a = a \% b$	Este operador asigna módulo al operando del lado izquierdo por el de la derecha y asigna el resultado a la variable.



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

3. Operadores de asignación 2.0

El siguiente video muestra en java como se implementan los operadores de asignación



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

4. Operadores Unarios

La siguiente tabla resume los operadores unarios en Java

Operadores incremento y decremento	
++	Incremento en 1 del operando.
--	Decremento en 1 del operando.



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

4. Operadores Unarios

El operador de incremento aumenta el valor de su operando en 1. El operando debe ser un tipo de dato aritmético o puntero, y debe hacer referencia a un objeto modificable. Similar mente, el operador de decremento disminuye el valor de su operando en 1.

Operador	Descripción	Ejemplo de expresión	Resultado del ejemplo
++	Incremento	4++	5
	i++ primero se utiliza la variable y luego se incrementa su valor	a=5; b=a++;	a vale 6 y b vale 5
--	decremento	4--	3
	++i primero se incrementa el valor de la variable y luego se utiliza	a=5; b=++a;	a vale 6 y b vale 6



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

4. Operadores Unarios

En el siguiente video se ejemplifica los operadores unarios en Java.



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

5. Operadores de igualdad y relacionales

La tabla a continuación resume los operadores de igualdad y relacionales.

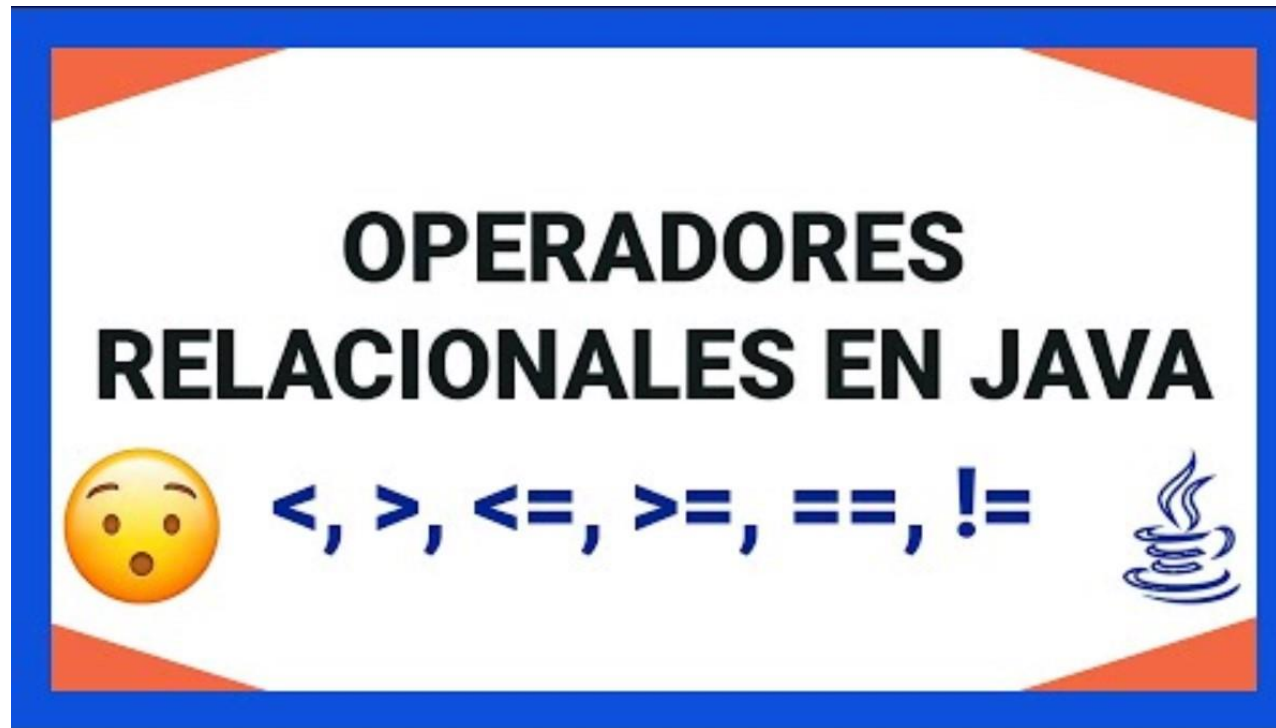
Operadores relacionales	
>	Mayor.
<	Menor.
>=	Mayor o igual.
<=	Menor o igual.
==	Igual.
!=	Distinto.
=	Asignación.



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

5. Operadores de igualdad y relacionales

En siguiente video se ilustra los operadores de igualdad y relacionales en Java.



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

6. Operadores lógicos

Este tipo de operadores, sirven para comparar dos condiciones. Se suelen utilizar en estructuras condicionales o el valor de dos o más condiciones.

&&	AND: Ambas condiciones deben cumplirse para entrar en el bloque SI.
	OR: Es suficiente con que una de las condiciones se cumpla para poder entrar en el bloque SI.



OPERADORES DE ASIGNACIÓN Y ARITMÉTICOS

6. Operadores lógicos

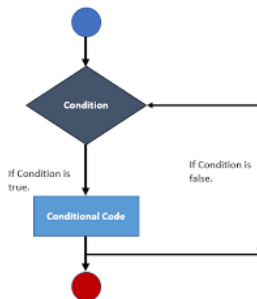
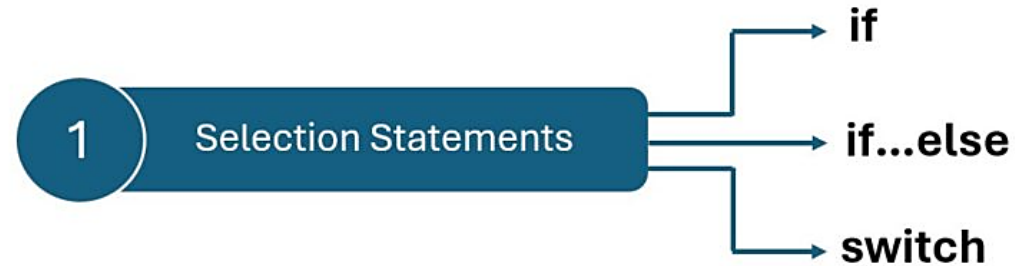
El video siguiente muestra en acción los operadores lógicos



ESTRUCTURAS CONDICIONALES

Estructura condicionales

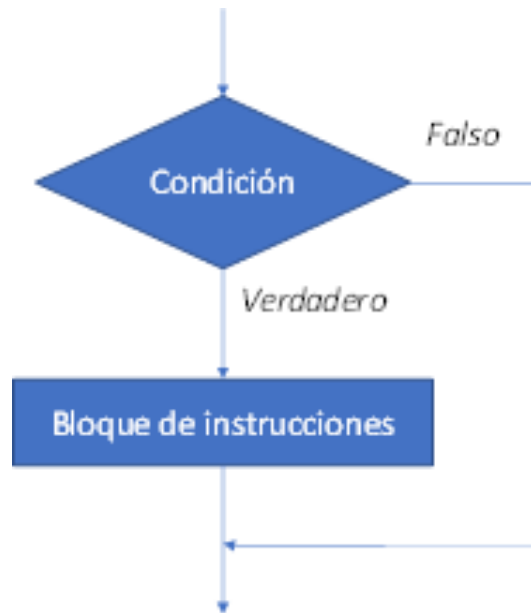
En java existen varias estructuras condicionales. A continuación se ilustra un resumen.



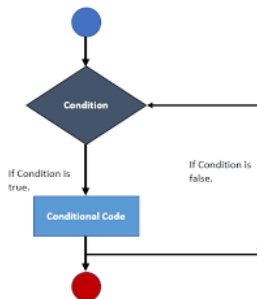
ESTRUCTURAS CONDICIONALES

Estructura if

Ejecuta un bloque de código si una condición especificada es verdadera.



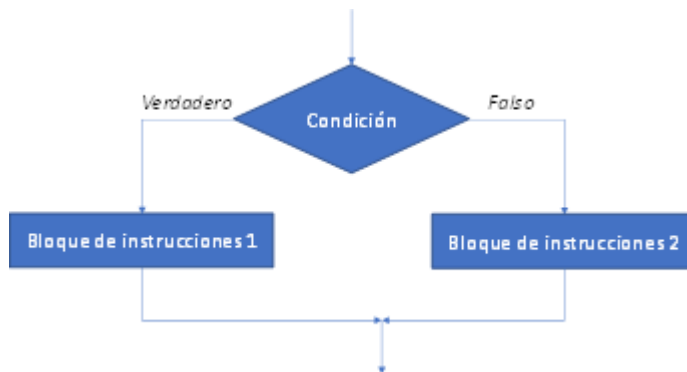
```
public static void main(String[] args) {  
    int edad = 18;  
    if (edad >= 18) {  
        System.out.println("Eres mayor de edad.");  
    } else {  
        System.out.println("Eres menor de edad.");  
    }  
}
```



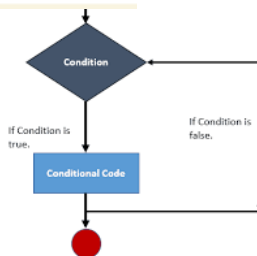
ESTRUCTURAS CONDICIONALES

Estructura if – else

En Java, las declaraciones if-else se utilizan para ejecutar diferentes bloques de código en función de si una condición es verdadera o falsa



```
public static void main(String[] args) {  
    int edad = 18;  
    if (edad >= 18) {  
        System.out.println("Eres mayor de edad.");  
    } else {  
        System.out.println("Eres menor de edad.");  
    }  
}
```

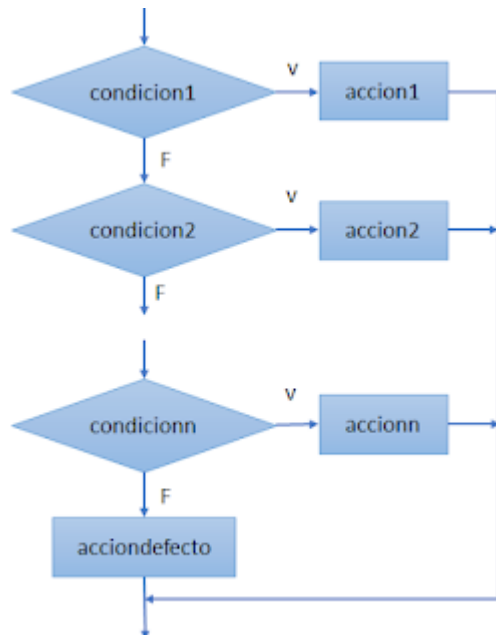


ESTRUCTURAS CONDICIONALES

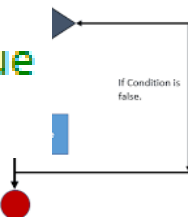
Estructura if – else-if

La estructura if-else-if en Java es una cadena de if, else if y, opcionalmente, un bloque else que proporciona una manera de ejecutar una de muchas rutas de código en función de múltiples condiciones.

Sintaxis



```
if (condition1) {  
    // Code to execute if condition1 is true  
} else if (condition2) {  
    // Code to execute if condition2 is true  
} else if (condition3) {  
    // Code to execute if condition3 is true  
} ...  
else {  
    // Code to execute if none of the above conditions are true  
}
```

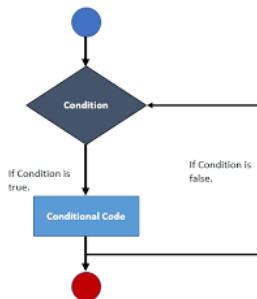


ESTRUCTURAS CONDICIONALES

Estructura if – else-if

Ejemplo


```
public static void main(String[] args) {  
    int calificacion = 75;  
    if (calificacion >= 90) {  
        System.out.println("Excelente trabajo, obtuviste una A.");  
    } else if (calificacion >= 80) {  
        System.out.println("Muy bien, obtuviste una B.");  
    } else if (calificacion >= 70) {  
        System.out.println("Buen esfuerzo, obtuviste una C.");  
    } else if (calificacion >= 60) {  
        System.out.println("Necesitas mejorar, obtuviste una D.");  
    } else {  
        System.out.println("Lo siento, has reprobado.");  
    }  
}
```



ESTRUCTURAS CONDICIONALES

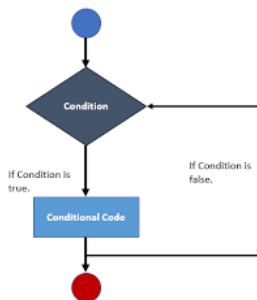

Estructura if – else-if

Video



SENTENCIA IF-ELSE

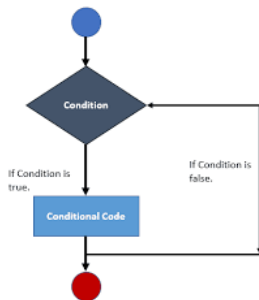
```
if(dinero <= 0) {  
    System.out.println("Estás arruinado!");  
} else {  
    System.out.println("NO estás arruinado!");  
}
```



ESTRUCTURAS CONDICIONALES

Estructura switch

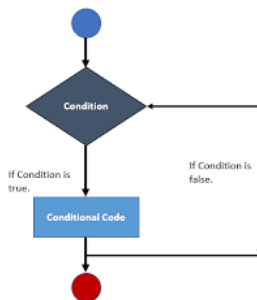
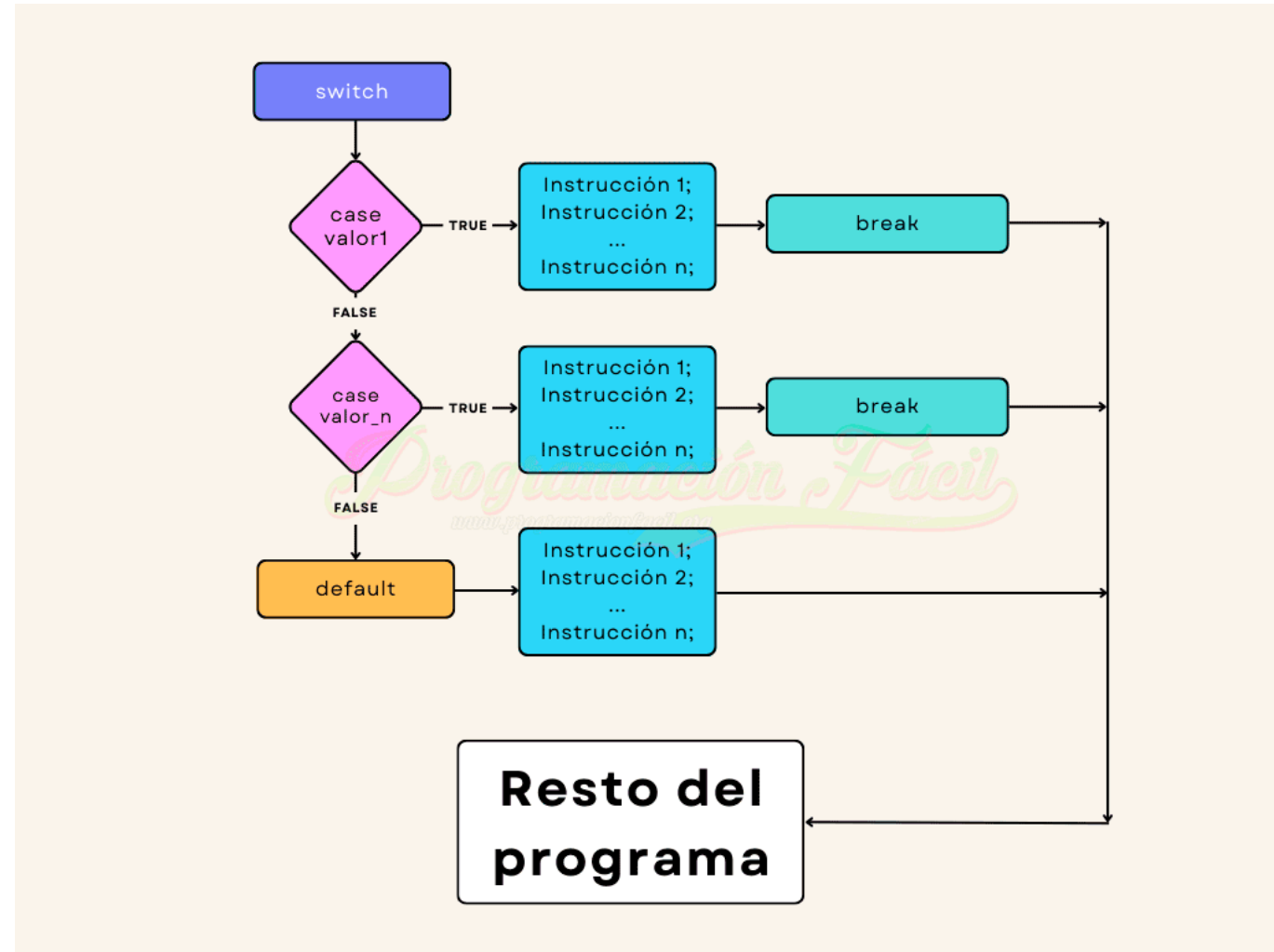
Permite evaluar una expresión y ejecutar diferentes bloques de código en función del valor de esa expresión.



ESTRUCTURAS CONDICIONALES

Estructura switch

Sintaxis Gráfica

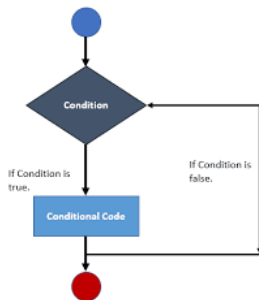


ESTRUCTURAS CONDICIONALES

Estructura switch

Sintaxis Java

```
switch (expresión) {  
    case valor1:  
        // instrucciones de código  
        break;  
    case valor2:  
        // instrucciones de código  
        break;  
    ...  
    case valor_n:  
        // instrucciones de código  
        break;  
    default:  
        // instrucciones de código  
}
```

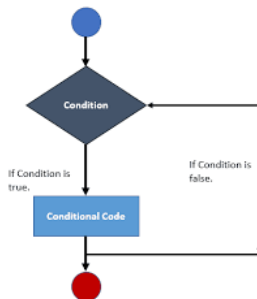


ESTRUCTURAS CONDICIONALES

Estructura switch

Ejemplo

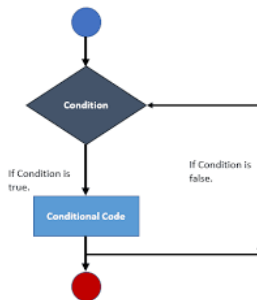
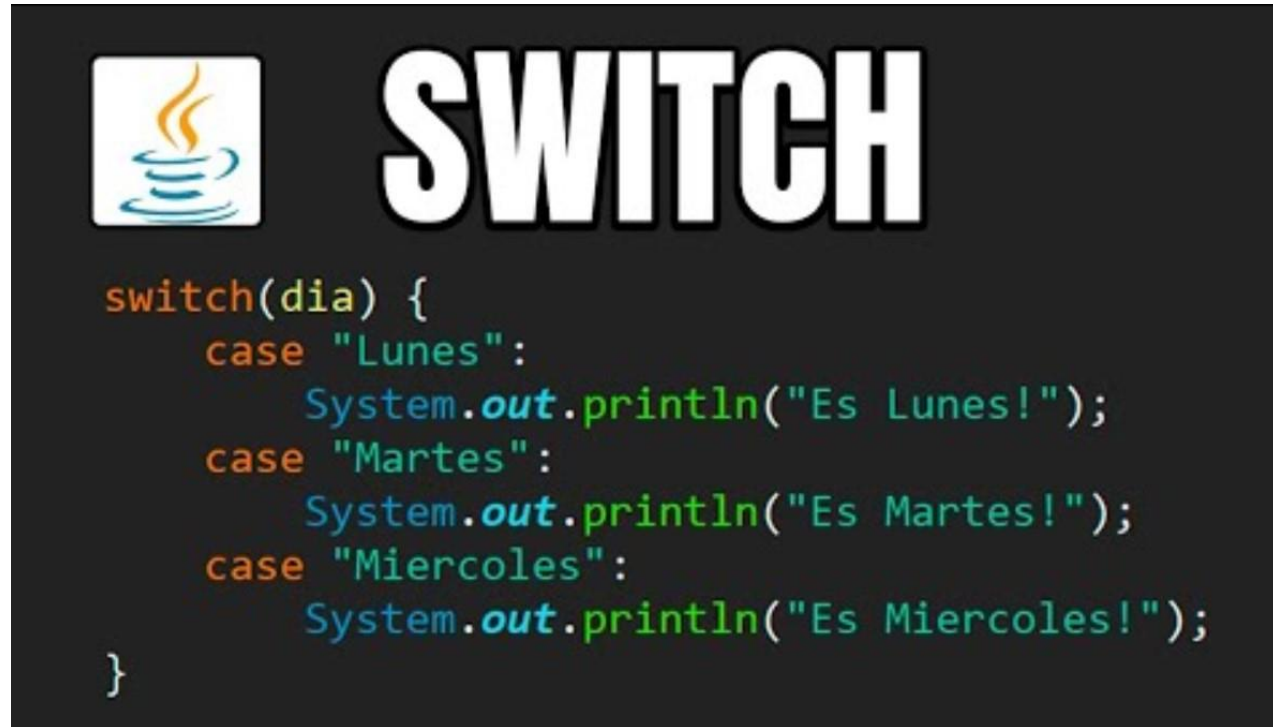
```
public static void main(String[] args) {  
    int error = 200;  
    //condicional switch  
    switch (error) {  
        case 200:  
            System.out.println("Todo ok.");  
            break;  
        case 301:  
            System.out.println("Movimiento permanente de la página.");  
            break;  
        case 302:  
            System.out.println("Movimiento temporal de la página.");  
            break;  
        case 404:  
            System.out.println("Página no encontrada.");  
            break;  
        case 500:  
            System.out.println("Error interno del servidor.");  
            break;  
        case 503:  
            System.out.println("Servicio no disponible.");  
            break;  
        default:  
            System.out.println("Código de error no disponible.");  
    }  
}
```



ESTRUCTURAS CONDICIONALES

Estructura switch

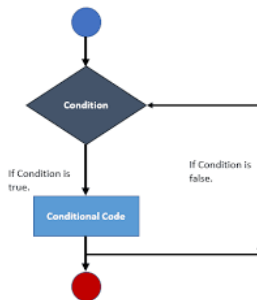
Video



ESTRUCTURAS CONDICIONALES

Estructura operador condicional ternario

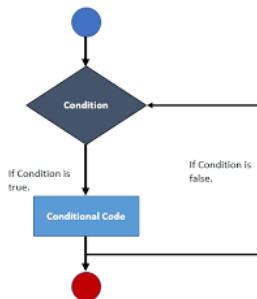
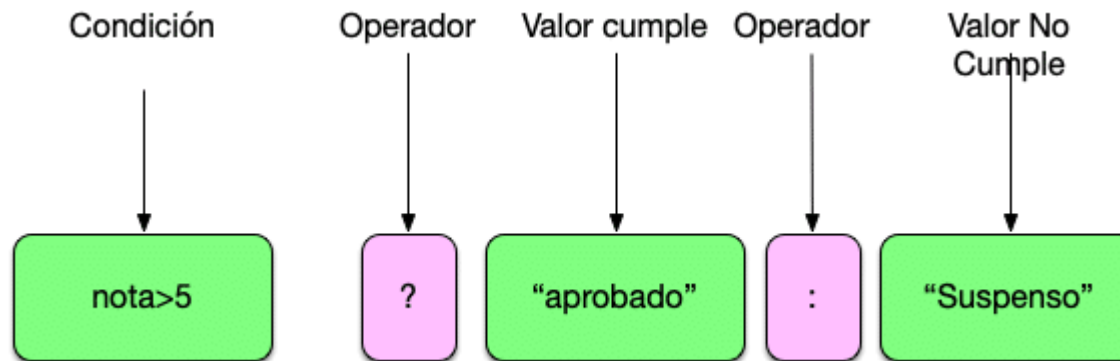
Normalmente este operador substituye a una estructura if /else es decir podemos tener un programa tan sencillo como el que valora si un alumno ha aprobado un examen o lo ha suspendido.



ESTRUCTURAS CONDICIONALES

Estructura operador condicional ternario

Sintaxis gráfica



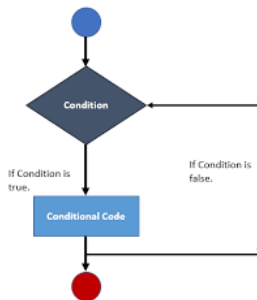
ESTRUCTURAS CONDICIONALES

Estructura operador condicional ternario

Sintaxis java

Operador Condicional Ternário

```
condition ? expr1 : expr2
```

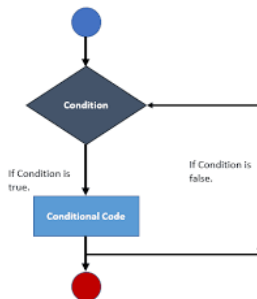


ESTRUCTURAS CONDICIONALES

Estructura operador condicional ternario

Ejemplo java

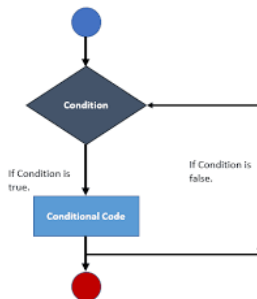
```
public static void main(String[] args) {  
    int aprobados = 1;  
    // Condicional usando IF - ELSE  
    if (aprobados == 1) {  
        System.out.println("ha aprobado " + aprobados + " un alumno");  
    } else {  
        System.out.println("han aprobado " + aprobados + " alumnos");  
    }  
  
    // Condicional usando el OPERADOR TERNARIO CONDICIONAL  
    System.out.println((aprobados == 1 ? "ha" : "han")  
        + " aprobado " + aprobados  
        + (aprobados == 1 ? " un alumno" : " alumnos"));  
}
```



ESTRUCTURAS CONDICIONALES

Estructura operador condicional ternario

Video



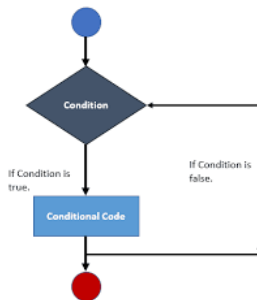
EJERCICIOS

Ejercicio # 1

Haga un programa en Java para haga el cálculo de valores resistivos en paralelo. Se lee tres valores ingresados por teclado.

La fórmula es la siguiente

$$Requivalente = \frac{1}{\frac{1}{r1} + \frac{1}{r2} + \frac{1}{r3}}$$



EJERCICIOS

Ejercicio # 2

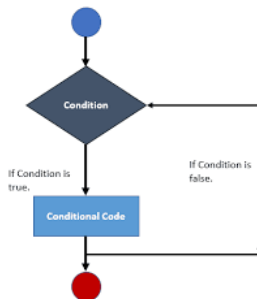
Dada la siguiente información de un usuario del servicio de agua:

- Documento de identidad
- Estado (A=activo, S=Suspendido)
- Estrato(1,2,3,4,5)

Se pide calcular el valor a pagar por tarifa básica, de acuerdo con las siguientes observaciones:

- Si el usuario está suspendido, el valor de tarifa básica es 0.
- Si el usuario está activo, el valor de la tarifa básica depende del estrato:

Estrato	Tarifa Básica
1	\$10.000
2	\$15.000
3	\$30.000
4	\$50.000
5	\$65.000



EJERCICIOS

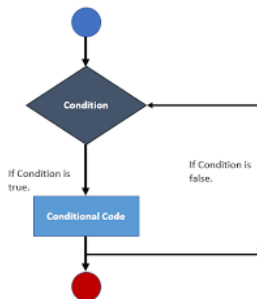
Ejercicio # 3

Escriba un programa en Java que lea un carácter y un número real, y que calcule el número real multiplicado por el factor de escala obtenido a partir del carácter, según los valores de la tabla:

Carácter	Factor de escala
'n'	0,000000001
'u'	0,000001
'm'	0,001
'K'	1000,0
'M'	1000000,0
'G'	1000000000,0

Use la instrucción *switch* para solucionar el problema.

Si se lee otro carácter que no esté en la tabla imprima *Double.NaN* para indicar el error.



EJERCICIOS

Ejercicio # 4

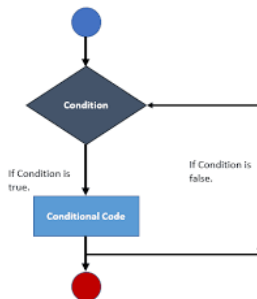
Escribe un programa en Python que determine si un año ingresado por el usuario es bisiesto o no. Un año bisiesto es aquel que es divisible entre 4, excepto aquellos divisibles entre 100 pero no entre 400.

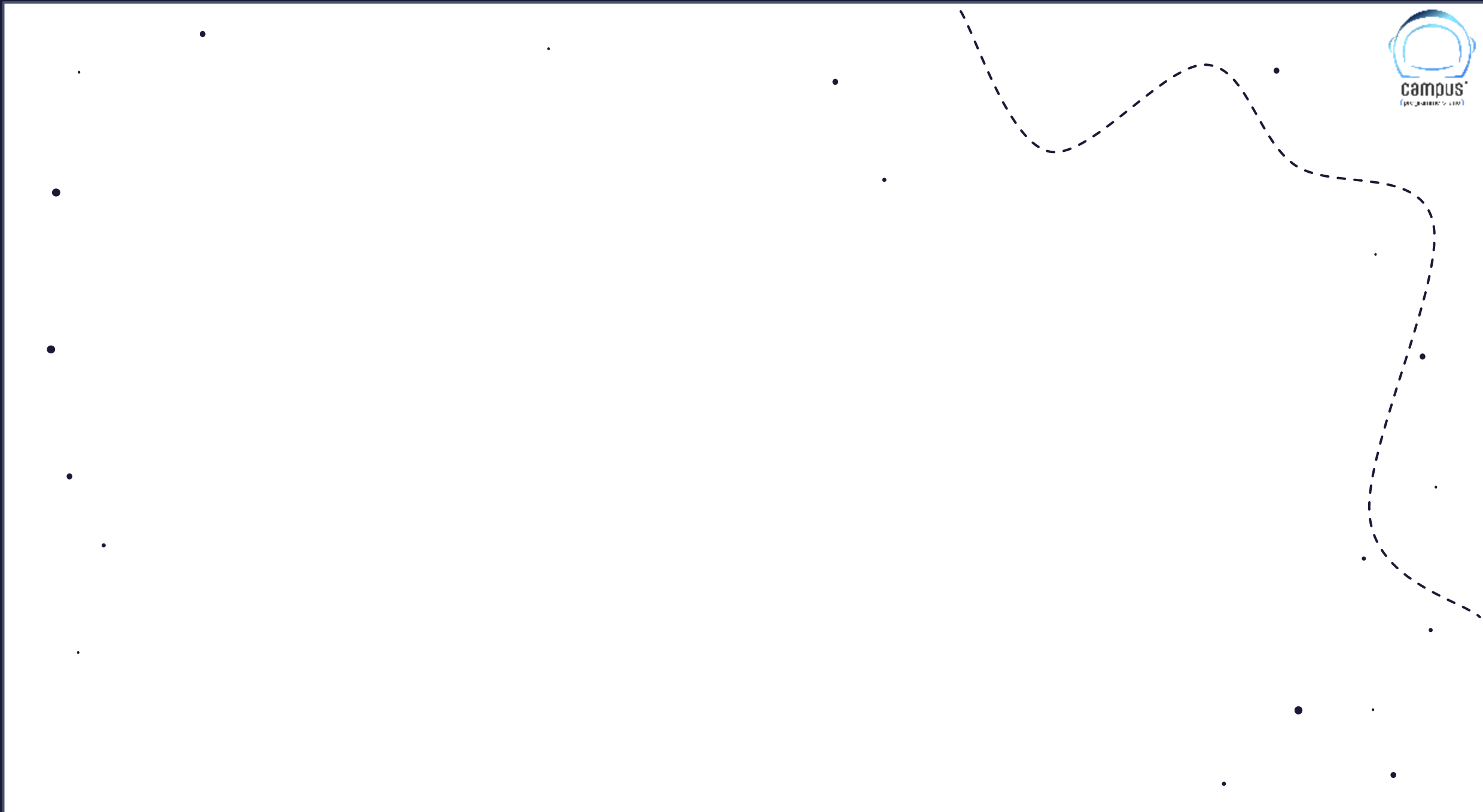
El programa debe realizar lo siguiente:

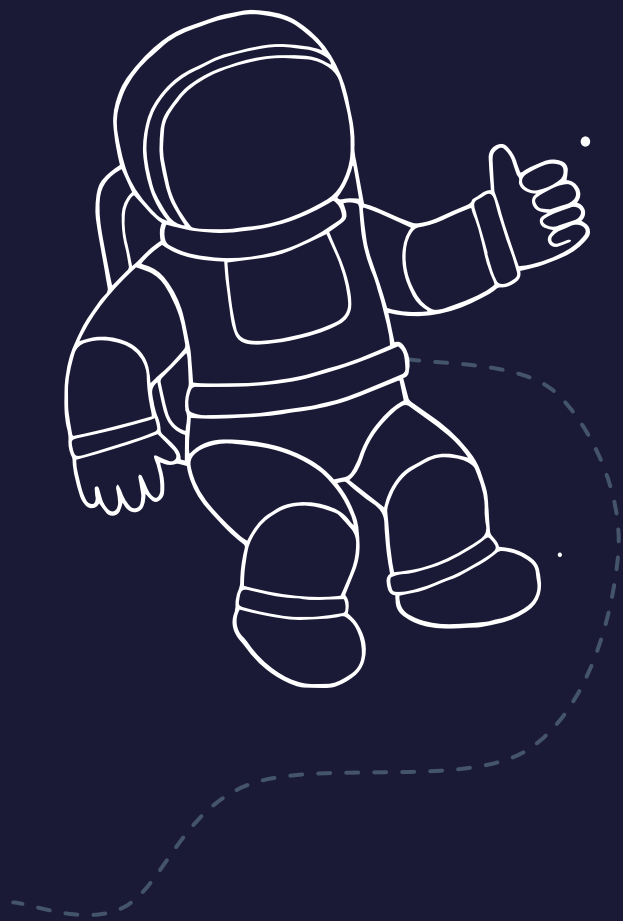
Solicitar al usuario que ingrese un año.

Verificar si el año cumple con las condiciones para ser bisiesto.

Mostrar un mensaje indicando si el año es bisiesto o no.







Programa académico CAMPUS

Ciclo 2

