



**UNIVERSIDAD
DE ANTIOQUIA**





LÓGICA Y REPRESENTACIÓN I

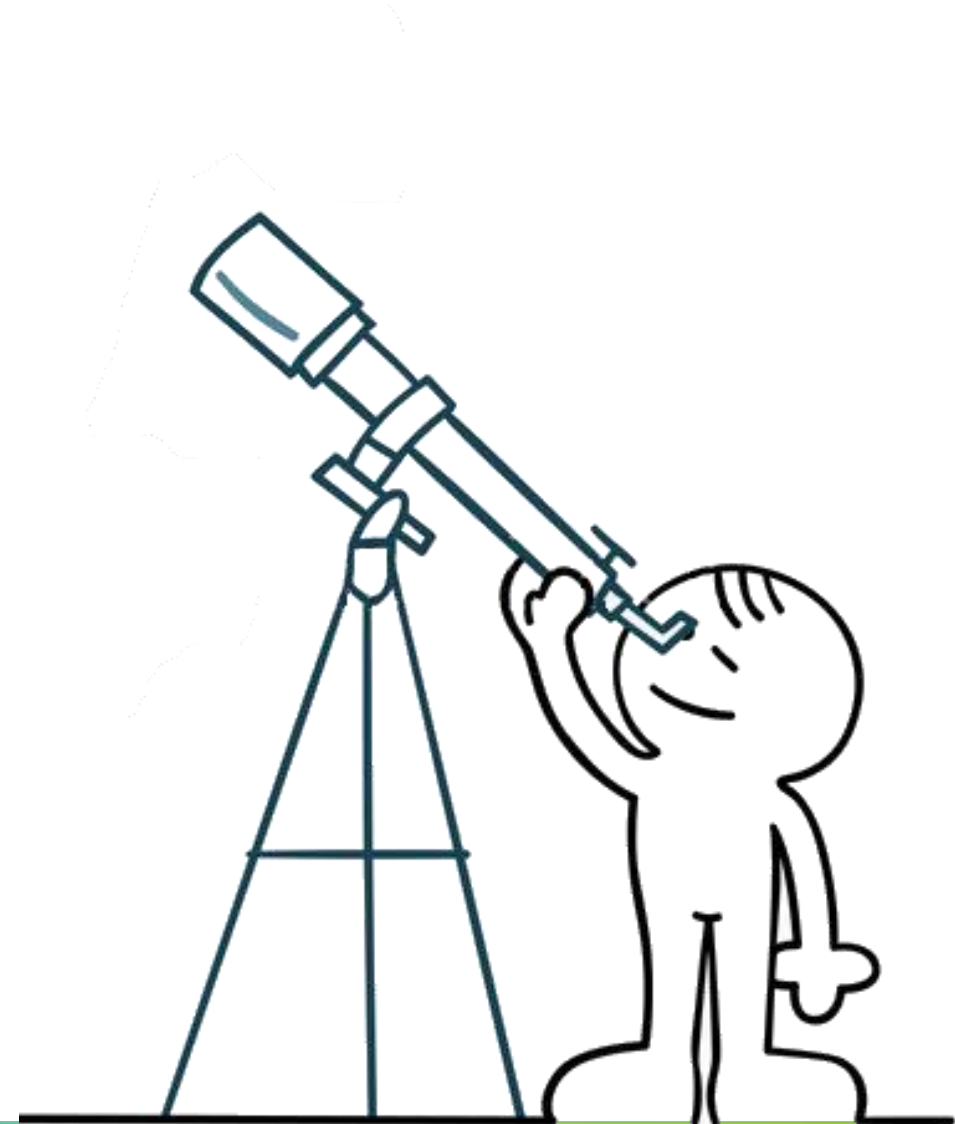
PROGRAMA DE INGENIERÍA DE SISTEMAS

Material desarrollado por **Carlos Andrés Mera Banguero**

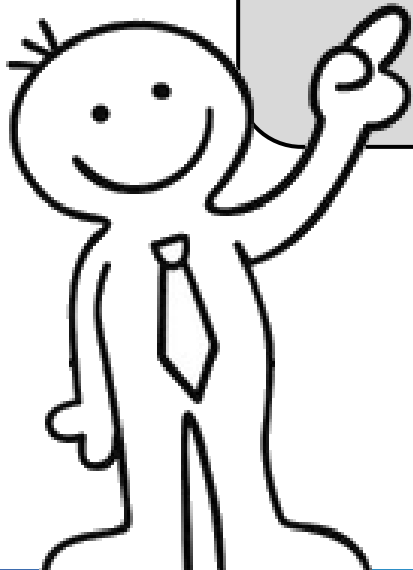
<https://github.com/carlosmera20/>

ESTRUCTURAS DE CONTROL

-  Condicional Simple
-  Condicional Doble
-  Condicional Anidado
-  Condicional Múltiple

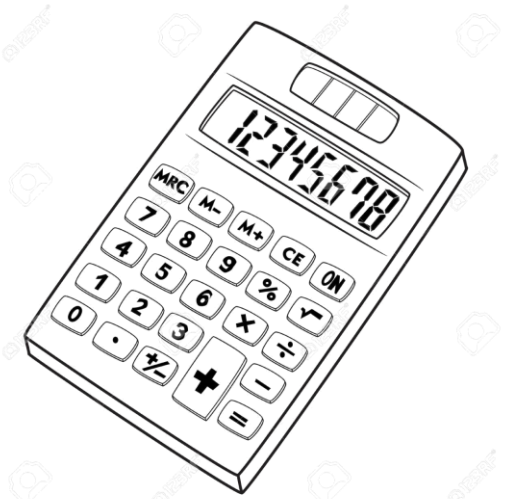


INTRODUCCIÓN





EJERCICIO: El profesor de matemáticas básicas requiere que usted programe una calculadora con las 4 operaciones básicas para que sus estudiantes puedan usarla desde el celular. Nota: la calculadora es básica así que solo opera de a dos números.



🦋 **ANÁLISIS DEL PROBLEMA:** Identifique el cliente, usuario, los requisitos funcionales, las entidades del mundo, las entradas, salidas y los procesos asociados al problema.

CLIENTE Y USUARIO

-
-



ENTIDADES DEL MUNDO

-
-



REQUERIMIENTOS FUNCIONALES

-
-



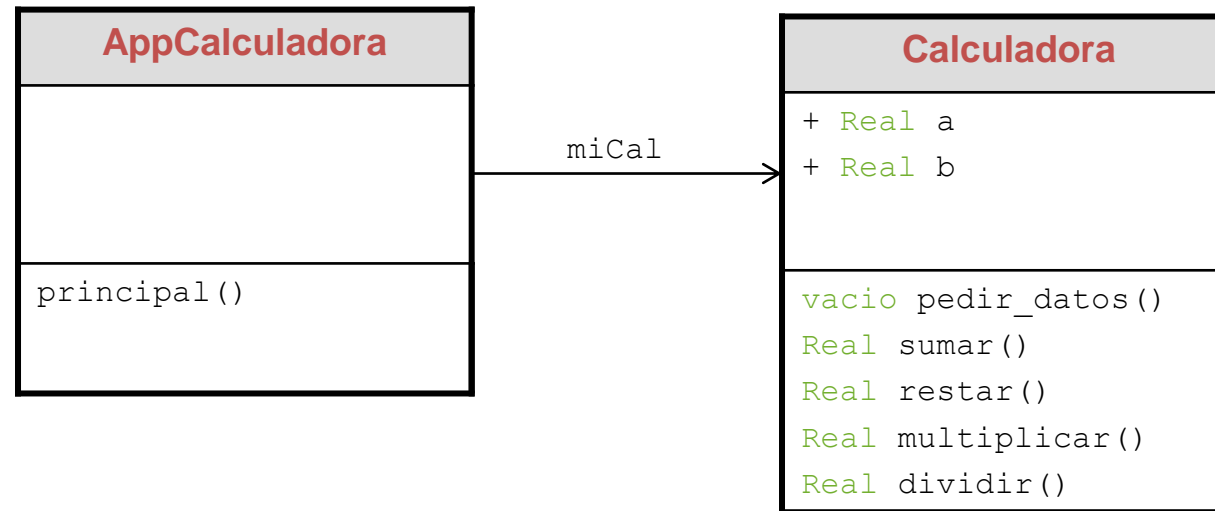
ENTRADAS, SALIDAS Y EL PROCESO

-
-



CLIENTE	El profesor de matemáticas básicas
USUARIO	Los estudiantes de matemáticas básicas
REQUERIMIENTOS FUNCIONALES	<p>El sistema debe permitir:</p> <ul style="list-style-type: none">▪ Sumar dos números▪ Restar dos números▪ Multiplicar dos números▪ Dividir dos números
ENTIDADES DEL MUNDO	<ul style="list-style-type: none">▪ Calculadora: tiene dos atributos reales (n_1 y n_2) que representan los números que se van a operar
ENTRADAS	<ul style="list-style-type: none">▪ Dos números que serán operados, los cuales deben ser reales
SALIDAS	<ul style="list-style-type: none">▪ Un número que es el resultado de una de las cuatro operaciones
PROCESO	<ul style="list-style-type: none">▪ Se piden los dos números a y b que será operados▪ Se calcula la suma como $c = a + b$▪ Se calcula la resta como $c = a - b$▪ Se calcula la multiplicación como $c = a * b$▪ Se calcula la división como $c = a / b$▪ Se muestra el resultado almacenado en c

- ✂ **DISEÑO DE LA SOLUCIÓN:** una vez identificadas las entidades definimos las clases con sus atributos y métodos, además de pintar las relaciones en las clases a desarrollar.



🦋 **DISEÑO DE LA SOLUCIÓN:** ahora se hacen los algoritmos de las clases:

Clase Calculadora

```
publico Real a, b #Atributos de la clase
```

```
publico vacio pedir_datos()
```

```
    Escribir("Ingrese el primer número: ")
```

```
    Leer(a)
```

```
    Escribir("Ingrese el segundo número: ")
```

```
    Leer(b)
```

```
Fin_Metodo
```

```
publico Real sumar()
```

```
    Real c
```

```
    c = a + b
```

```
    Retornar c
```

```
Fin_metodo
```

```
publico Real restar()
```

```
    Real c
```

```
    c = a - b
```

```
    Retornar c
```

```
Fin_metodo
```

```
publico Real multiplicar()
```

```
    Real c
```

```
    c = a * b
```

```
    Retornar c
```

```
Fin_metodo
```

```
publico Real dividir()
```

```
    Real c
```

```
    c = a / b
```

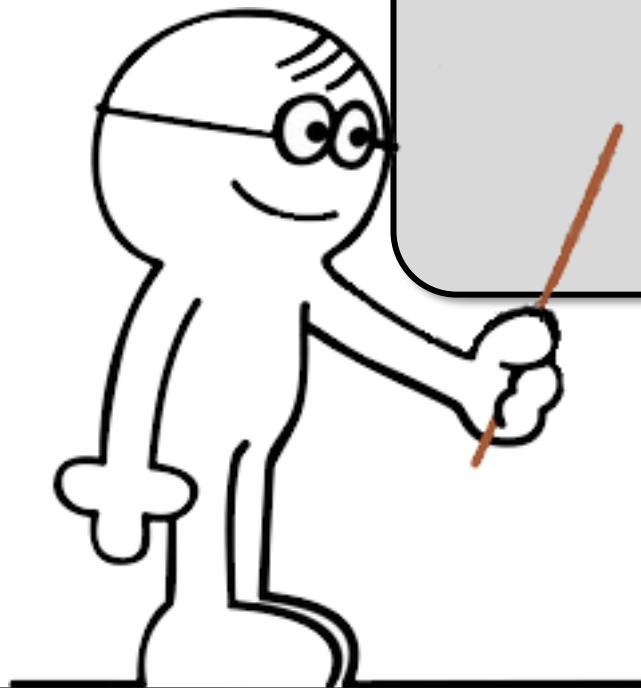
```
    Retornar c
```

```
Fin_metodo
```

```
Fin_clase
```



¿Identifica cuál es el error
en este método?



ESTRUCTURAS CONTROL

ESTRUCTURAS DE CONTROL

Las estructuras de control permiten cambiar el flujo de ejecución de un programa, haciendo que ciertos bloques de código se ejecuten si y solo si se dan unas condiciones particulares.



SELECTIVAS



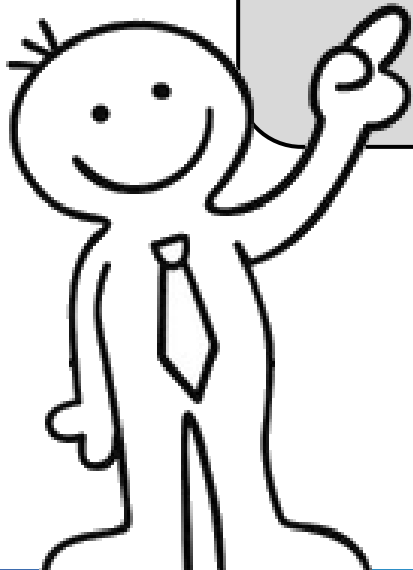
Que permiten determinar si un bloque se ejecuta o no según si una **condición** es verdadera o falsa.

REPETITIVAS



Que permiten que un bloque de código se ejecuta múltiples veces según si una **condición** es verdadera o falsa.

ESTRUCTURAS DE CONTROL SELECTIVAS:
CONDICIONAL SIMPLE



ESTRUCTURAS DE CONTROL SELECTIVAS



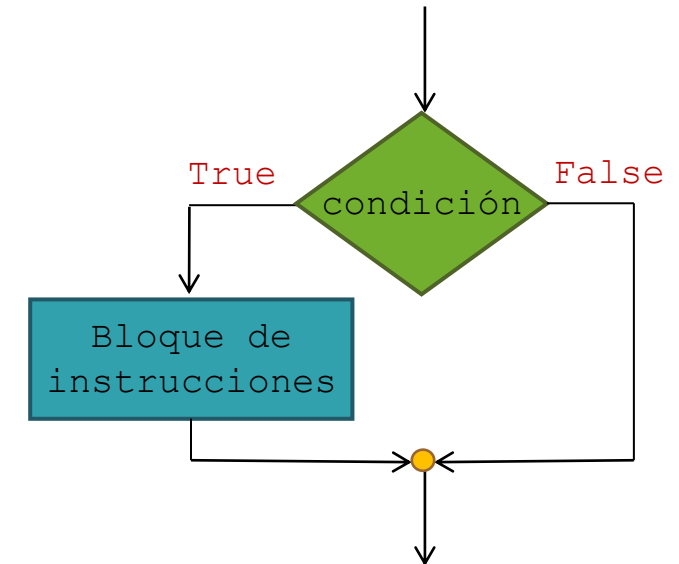
UNIVERSIDAD
DE ANTIOQUIA

CONDICIONAL SIMPLE:

- Representa la forma más sencilla de tomar una decisión en el programa.

CONDICIONAL SIMPLE

En general, los condicionales permiten al programa tomar decisiones y modificar el flujo de ejecución del mismo. Específicamente, el condicional simple evalúa una operación lógica (llamada **condición**) y ejecuta el bloque de instrucciones de código siguiente cuando el resultado de la operación lógica es **verdadero**.



ESTRUCTURAS DE CONTROL SELECTIVAS



UNIVERSIDAD
DE ANTIOQUIA

CONDICIONAL SIMPLE:

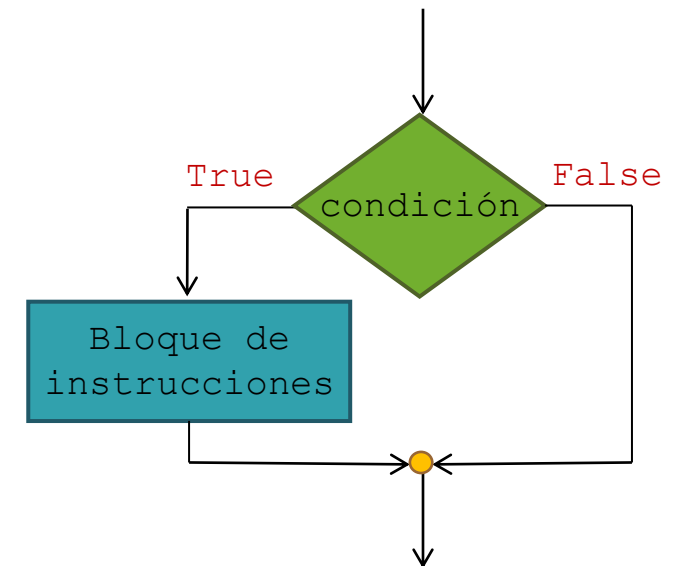
- Representa la forma más sencilla de tomar una decisión en el programa.

Algoritmo	Python
<pre>Si (condición) Entonces instrucción_1 instrucción_2 ... Fin_Si</pre>	<pre>if (condición): instrucción_1 instrucción_2 ...</pre>



La **condición** es una expresión lógica

El **bloque de instrucciones** se ejecuta sólo cuando la condición es **verdadera**.





EJERCICIO: El profesor de matemáticas también le ha pedido que agregue a la calculadora una funcionalidad que le indique a los estudiantes si cada uno de los números ingresados es positivo. En caso de serlo debe mostrar el mensaje: `El numero xx es positivo!`



ESTRUCTURAS DE CONTROL SELECTIVAS



UNIVERSIDAD
DE ANTIOQUIA

🚀 **ALGORITMO:** para dar solución al problema agregamos un método que muestre un mensaje indicando si esos números son positivos.

Algoritmo

```
publico vacio verificar_positivos()

    Si (a > 0) Entonces
        Escribir("El número ",a," es positivo!")
    Fin_Si

    Si (b > 0) Entonces
        Escribir("El número ",b," es positivo!")
    Fin_Si

Fin_Metodo
```

Python

```
def verificar_positivos():

    if (a > 0):
        print(f"El número {a} es positivo!")

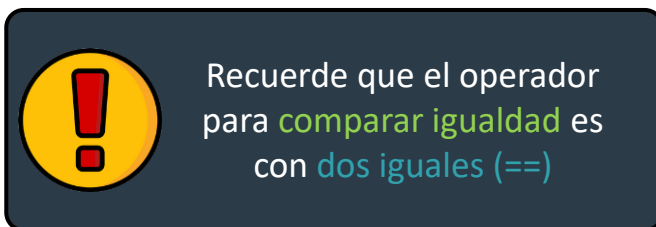
    if (n2 > 0):
        print(f"El número {b} es positivo!")
```



EJERCICIO: Ahora haga un método que verifique si los dos números ingresados en la calculadora son iguales, de serlo deben mostrar el mensaje `Los números son iguales`



🌟 **ALGORITMO:** para dar solución al problema agregamos un método que muestre un mensaje indicando si esos números son positivos.



```
publico vacio verificar_si_son_iguales()
```

```
Si (a == b) entonces
```

```
    Escribir("Los dos números son iguales")
```

```
Fin_Si
```

```
Fin_Metodo
```

🏆 **ALGORITMO:** para dar solución al problema agregamos un método que muestre un mensaje indicando si esos números son positivos.

Algoritmo

```
publico vacio verificar_si_son_iguales()  
  
    Si (a == b) entonces  
        Escribir("Los dos números son iguales")  
    Fin_Si  
  
Fin_Metodo
```

Python

```
def verificar_si_son_iguales():  
  
    if (a == b):  
        print("Los dos números son iguales")
```



ESTRUCTURAS CONTROL SELECTIVAS:
CONDICIONAL DOBLE

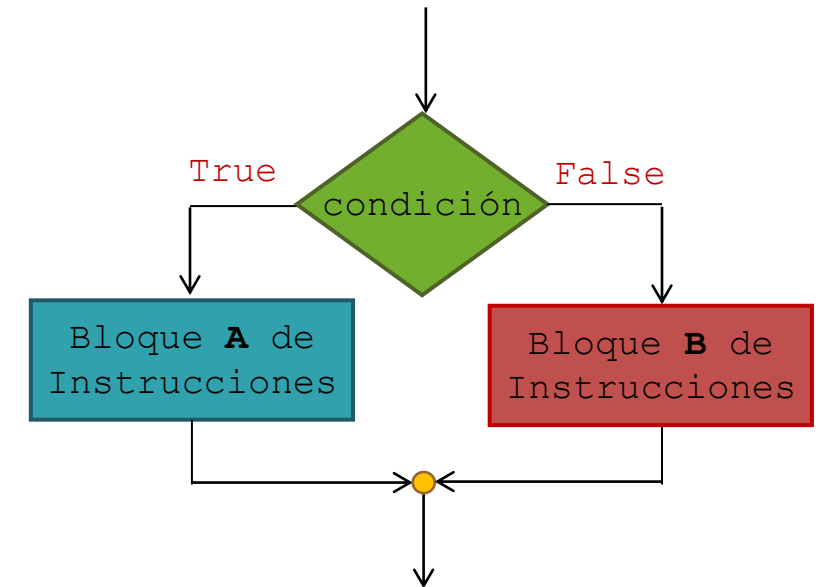


CONDICIONAL DOBLE:

- Este tipo de condicionales considera una bifurcación de dos caminos en una decisión dentro del programa. Los caminos son excluyentes, es decir, se toma uno u otro, nunca se pueden tomar ambos caminos al tiempo.

CONDICIONAL DOBLE

A diferencia del condicional simple, el condicional doble tiene dos caminos de acción: uno cuando la condición es **verdadera** y otro cuando la condición es **falsa**.



ESTRUCTURAS DE CONTROL SELECTIVAS

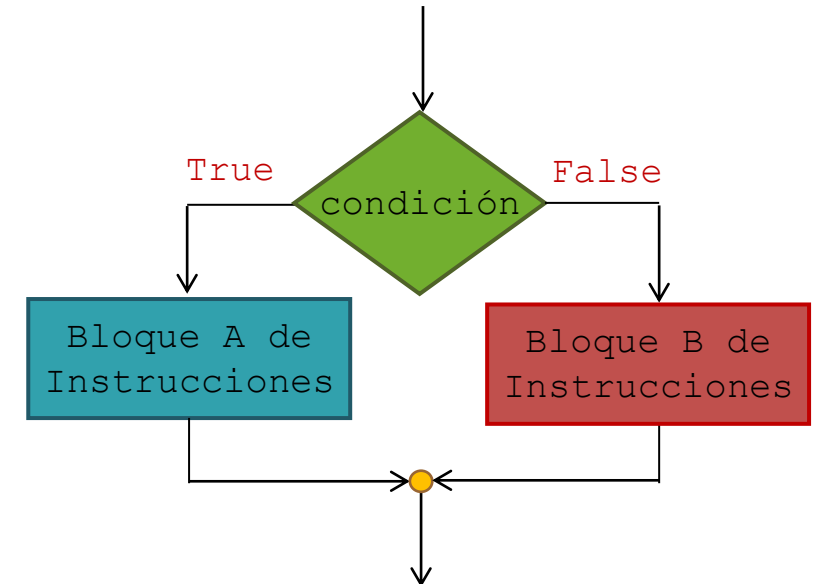


UNIVERSIDAD
DE ANTIOQUIA

CONDICIONAL DOBLE:

- Considera una bifurcación de dos caminos en una decisión dentro del programa. Los caminos son excluyentes, es decir, se toma uno u otro, nunca se pueden tomar ambos caminos al tiempo.

Algoritmo	Python
<pre>Si (condición) Entonces Bloque A de Instrucciones Sino Bloque B de Instrucciones Fin_Si</pre>	<pre>if (condición): Bloque A de Instrucciones else: Bloque B de Instrucciones</pre>



La **condición** es una expresión lógica

El **Bloque A** de instrucciones se ejecuta sólo cuando la condición es **verdadera**.

El **Bloque B** de instrucciones se ejecuta sólo cuando la condición es **falsa**.



EJERCICIO: Mejore el método del ejercicio anterior para que muestre un mensaje indicando si los números son iguales o diferentes.



🦋 **ALGORITMO:** Mejore el método del ejercicio anterior para que muestre en pantalla si los números son iguales o si son diferentes

Algoritmo

```
publico vacio verificar_si_son_iguales()  
  
    Si (a == b) entonces  
        Escribir("Los dos números son iguales")  
    Sino  
        Escribir("Los dos números son diferentes")  
    Fin_Si  
  
Fin_Metodo
```

Python

```
def verificar_si_son_iguales():  
  
    if (a == b):  
        print("Los dos números son iguales")  
    else:  
        print("Los dos números son diferentes")
```



EJERCICIO: Ahora el profesor de matemáticas le ha pedido que desarrolle un método que indique si el primero número ingresado a la calculadora es par o no.



A cartoon drawing of a teacher with glasses, holding a pointer stick and pointing at a large grey box containing a question.

¿CÓMO SE SABE QUE UN NÚMERO ES PAR?

🚀 **ALGORITMO:** Desarrolle un método que indique si el primer número ingresado en la calculadora es par o no

Algoritmo

```
publico vacío verificar_si_es_par()
```

```
Si (a MOD 2 == 0) entonces
```

```
    Escribir("El primer número ingresado es par")
```

```
Sino
```

```
    Escribir("El primer número ingresado NO es par")
```

```
Fin_Si
```

```
Fin_Metodo
```

Python

```
def verificar_si_es_par():
```

```
    if (a % 2 == 0):
```

```
        print("El primer número ingresado es par")
```

```
    else:
```

```
        print("El primer número ingresado NO es par")
```



EJERCICIO: En un estacionamiento se cobra \$5.400 por cada hora de parqueo, más \$1.350 por cada fracción de 15 minutos adicional. Diseñe un algoritmo que determine cuanto debe pagar un cliente por el estacionamiento de su vehículo, conociendo la hora de entrada y la hora de salida del vehículo, ambas en formato hh:mm.





ESTRUCTURAS CONTROL:
CONDICIONALES ANIDADOS

ESTRUCTURAS DE CONTROL SELECTIVAS



UNIVERSIDAD
DE ANTIOQUIA

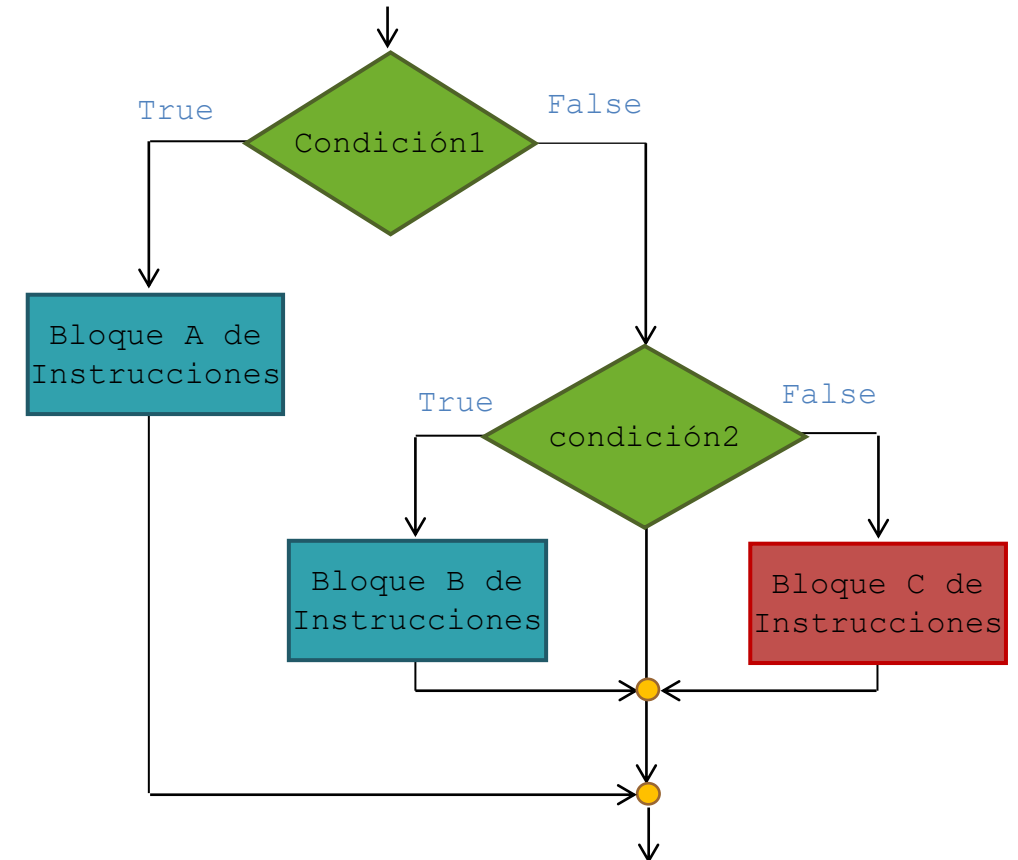


CONDICIONAL ANIDADO:

- Es la estructura más compleja entre los condicionales y se usan para tomar decisiones, dentro de otras decisiones ya tomadas.

CONDICIONAL ANIDADO

Decimos que una estructura *condicional* es *anidada* cuando dentro de una de las ramas de un condicional hay otra estructura *condicional*.



ESTRUCTURAS DE CONTROL SELECTIVAS

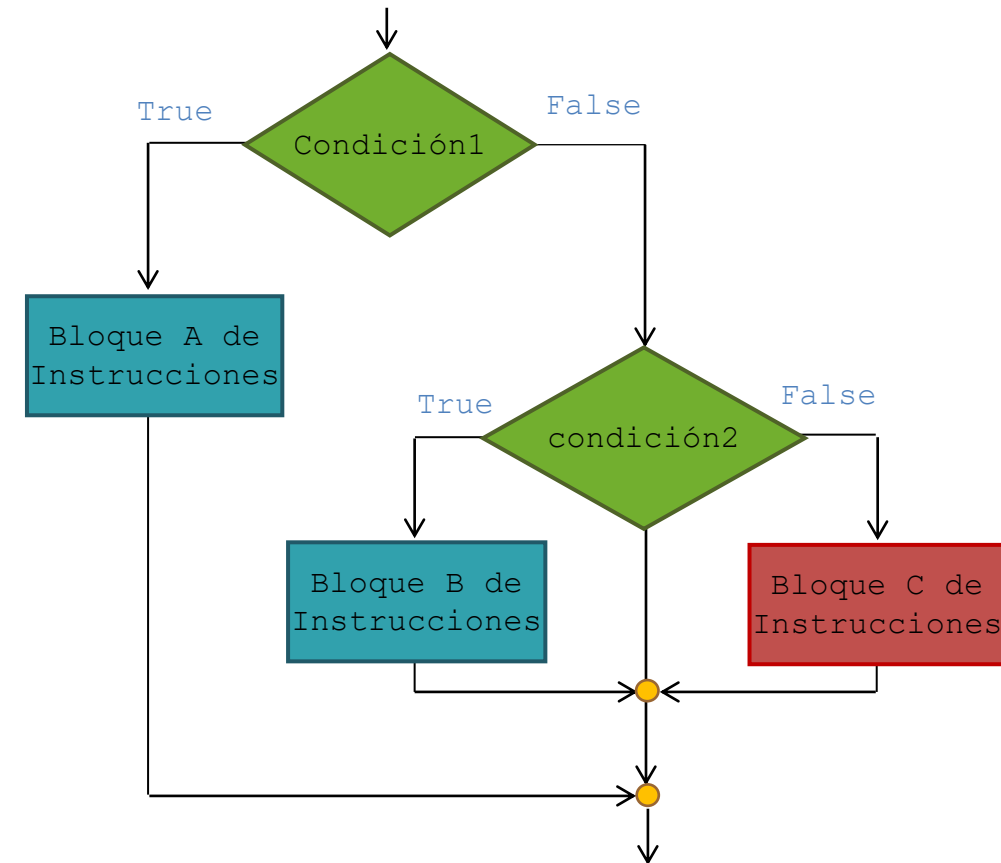


UNIVERSIDAD
DE ANTIOQUIA

CONDICIONAL ANIDADO:

Es la estructura más compleja entre los condicionales y se usan para tomar decisiones, dentro de otras decisiones ya tomadas.

Algoritmo	Python
<pre>Si (condición1) entonces Bloque A de Instrucciones Sino Si (condición2) entonces Bloque B de Instrucciones Sino Bloque C de Instrucciones Fin_Si Fin_Si</pre>	<pre>if (condición1): Bloque A de Instrucciones elif (condición2): Bloque B de Instrucciones else: Bloque C de Instrucciones</pre>





EJERCICIO: Ahora haga un método llamado `verificar_numero` que verifique y muestre un mensaje en la pantalla indicando si el primer número de la calculadora es positivo, negativo o igual a cero



🚀 **ALGORITMO:** Desarrolle un algoritmo que pida un número y verifique si éste es positivo, negativo o igual a cero

Algoritmo

```
publico vacio verificar_numero()  
  
    Si (a == 0) entonces  
        Escribir("El número es igual a 0")  
    Sino  
        Si (a > 0) entonces  
            Escribir("El número es POSITIVO")  
        Sino  
            Escribir("El número es NEGATIVO")  
        Fin_Si  
    Fin_Si  
Fin_Metodo
```

Python

```
def verificar_numero():  
  
    if (a == 0):  
        print("El número es igual a 0")  
    elif (a > 0):  
        print("El número es POSITIVO")  
    else:  
        print("El número es NEGATIVO")
```




EJERCICIO: Desarrolle un método llamado **mostrar_el_mayor** en la calculadora que compare los dos números ingresados a la calculadora e indique si estos son iguales o muestre cual es el mayor



ESTRUCTURAS DE CONTROL SELECTIVAS



UNIVERSIDAD
DE ANTIOQUIA

🚀 **ALGORITMO:** Desarrolle un algoritmo que compare dos números e indique si estos son iguales, o si uno es mayor que el otro.

Algoritmo

```
publico vacio mostrar_el_mayor()

Si (n1 == n2) entonces
    Escribir("Los dos números en la calculadora son iguales")
Sino
    Si (n1 > n2) entonces
        Escribir(n1, " es mayor que ", n2)
    Sino
        Escribir(n2, " es mayor que ", n1)
Fin_Si
Fin_Si
Fin_Metodo
```

Python

```
def mostrar_el_mayor():

    Si (a == b):
        print("Los dos números en la calculadora son iguales")
    elif (n1 > n2):
        print(f"{a} es mayor que {b}")
    else:
        print(f"{b} es mayor que {a}")
```



EJERCICIO: Considere una clase **Persona** que tiene un atributo denominado **edad**: desarrolle un método en la clase persona que **retorne** una cadena indicando si la persona es un niño (con menos de 10 años), adolescente (entre 10 y 17 años), adulto joven (entre 18 y 25), adulto (entre 26 y 34), adulto contemporáneo (entre 35 y 55) o adulto mayor (más de 55).



ESTRUCTURAS DE CONTROL SELECTIVAS



UNIVERSIDAD
DE ANTIOQUIA

```
Clase Persona
publico Entero edad
publico Cadena obtener_tipo_persona()
    Si (edad > 0 AND edad < 10) entonces
        Retornar "La persona es considerada un niño"
    Sino
        Si (edad >= 10 AND edad <= 17) entonces
            Retornar "La persona es considerada un adolescente"
        Sino Si (edad >= 18 AND edad <= 25) entonces
            Retornar "La persona es considerada un adulto joven y ya puede votar!"
        Sino Si (edad >= 26 AND edad <= 34) entonces
            Retornar "La persona es considerada es un adulto en plenitud"
        Sino Si (edad >= 35 AND edad <= 55) entonces
            Retornar "La persona es considerada un adulto contemporáneo ..."
        Sino Si (edad >= 56) entonces
            Retornar "La persona es considerada un adulto mayor ..."
        Sino
            Retornar "Error: parece que la edad de la persona no es válida!"
        Fin_Si
    Fin_Si
Fin_Si
Fin_Si
Fin_Si
Fin_Si
Fin_Si
Fin_Metodo
Fin_Clase
```

🦋 **ALGORITMO:** El almacén “*Aprenda a vestirse*” ha iniciado su temporada de rebajas. Durante este periodo a las compras se le pueden aplicar diferentes descuentos así:

- Si el total de la compra es mayor o igual a 850.001 se le aplica un descuento del 40%.
- Si el total de la compra está entre 500.001 y 850.000 se le aplica un descuento del 30%.
- Si el total de la compra está entre 200.001 y 500.000, se le aplica un descuento del 20%.
- Para compras inferiores a 200.000 el descuento es del 5%.

Además, el almacén está promoviendo la política del buen vestir en los adultos mayores, por lo que está dando un descuento adicional que corresponde a una tercera parte de la edad del comprador más un 5%. Esto cuando la edad del cliente es mayor o igual a 60 años.

El programa debe mostrar el valor original de la compra, el valor de cada descuento aplicado y el total a pagar por el cliente.

✈ **ANÁLISIS DEL PROBLEMA:** Identifique el cliente, usuario, los requisitos funcionales, las entidades del mundo, las entradas, salidas y los procesos asociados al problema.

CLIENTE Y USUARIO

-
-



ENTIDADES DEL MUNDO

-
-



REQUERIMIENTOS FUNCIONALES

-
-



ENTRADAS, SALIDAS Y EL PROCESO

-
-



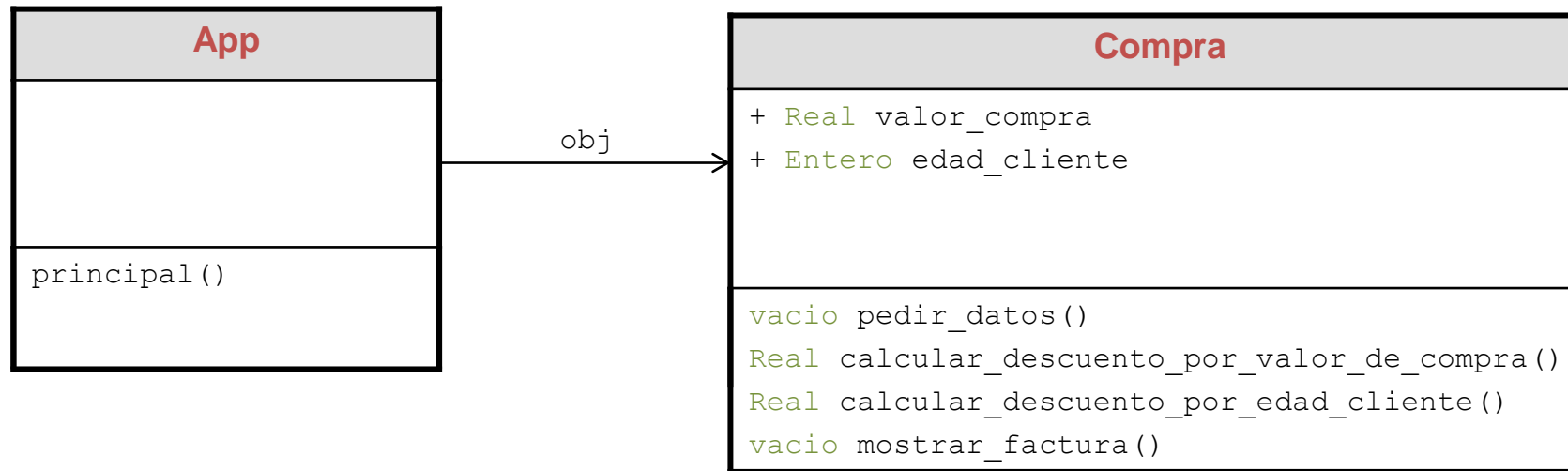
ESTRUCTURAS DE CONTROL SELECTIVAS



UNIVERSIDAD
DE ANTIOQUIA

CLIENTE	El almacén aprenda a vestirse
USUARIO	El cajero del almacén
REQUERIMIENTOS FUNCIONALES	<p>El sistema debe permitir:</p> <ul style="list-style-type: none">▪ Calcular un descuento por el valor de la compra▪ Calcular un descuento por la edad del comprador▪ Calcular el valor a pagar con base en los descuentos obtenidos
ENTIDADES DEL MUNDO	<ul style="list-style-type: none">▪ Compra: que representa a la compra que se está haciendo y que tiene como atributos, el valor de la compra y la edad del cliente
ENTRADAS	<ul style="list-style-type: none">▪ El valor de la compra, el cual es un número entero mayor a cero
SALIDAS	<ul style="list-style-type: none">▪ El valor aplicado por el descuento del valor de la compra▪ El valor aplicado por el descuento de la edad▪ El valor neto a pagar
PROCESO	<ul style="list-style-type: none">▪ Se pide el valor de la compra▪ Se calcula los descuentos siguiendo las siguientes fórmulas: $d1 = \text{valorCompra} * \text{porcentajeDescuento}$ $d2 = \text{valorCompra} * (\text{edadCliente}/3 + 5)/100$▪ El valor a pagar neto a pagar se calcula como: $\text{valorCompra} - d1 - d2$

🦋 **DISEÑO DE LA SOLUCIÓN:** una vez identificadas las entidades definimos las clases con sus atributos y métodos, además de pintar las relaciones en las clases a desarrollar.



🦋 **ALGORITMO:** Escribe un programa que permita predecir el estado del tiempo con base en la temperatura, el viento y la humedad.

La predicción debe usar las siguientes reglas:

- Si la temperatura $> 25^{\circ}\text{C}$, viento $< 10\text{ km/h}$, humedad $< 33\%$ \Rightarrow soleado
- Si la temperatura está entre 20 y 30 C, viento $< 10\text{ km/h}$, humedad $> 90\%$ \Rightarrow húmedo
- Si la temperatura está entre 0 y 10 C, viento $> 5\text{ km/h}$, humedad entre 40% y 80% \Rightarrow frío
- Si la temperatura está entre -5 y 0 C y el viento $< 5\text{ km/h}$ o humedad $> 50\%$ \Rightarrow nieve

El programa debe mostrar el estado de cada posible estado del clima, por ejemplo: El Día estará soleado pero no húmedo, ni frío, ni habrá nieve. Además, el programa debe funcionar para cualquier valor de temperatura, viento y humedad. Sin embargo, si no se cumple ninguna condición se debe indicar que el estado del clima es incierto.

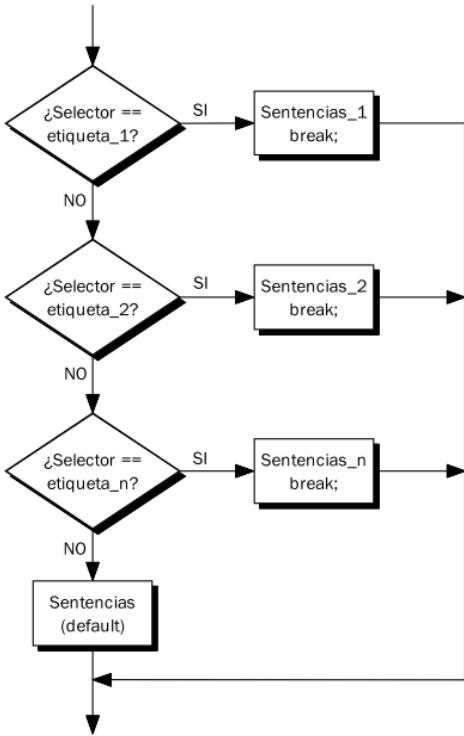


ESTRUCTURAS CONTROL:
CONDICIONAL MÚLTIPLE

CONDICIONALES MÚLTIPLES:

- Este tipo de condicional es utilizado para seleccionar una, entre muchas alternativas disponibles. Básicamente, esta expresión evalúa una *variable*, llamada variable de control, y con base en su valor selecciona el *caso* que debe ejecutarse. Si el valor de la variable no es igual a ninguno de los casos, se ejecutan las instrucciones del bloque *en_otro_caso*.

Algoritmo	Python
<pre>Segun_Sea(variable) haga caso valor1: instrucciones1 ... Interrumpir caso valor2: instrucciones2 ... Interrumpir caso valor3: instrucciones3 ... Interrumpir en_otro_caso: instruccionesOtras ... Fin Segun_sea</pre>	<p>NO EXISTE!</p> <p>Se emula con condicionales anidados!</p>





EJERCICIO: Haga una clase principal que lea un número entre 1 y 10 y muestre el número escrito en romanos.



ESTRUCTURAS DE CONTROL SELECTIVAS



UNIVERSIDAD
DE ANTIOQUIA

Algoritmo

```
publico vacio convertir_a_romano()  
    Entero a  
    Escribir("Ingrese un número del 1 al 10: ")  
    Leer(a)  
    Según_sea (a) haga  
        caso 1:  
            Escribir(a, " en romanos es: I")  
            Interrumpir  
        caso 2:  
            Escribir(a, " en romanos es: II")  
            Interrumpir  
        caso 3:  
            Escribir(a, " en romanos es: III")  
        ...  
        caso 10:  
            Escribir(a, " en romanos es: X")  
            Interrumpir  
    en_otro_caso  
        Escribir("El valor es desconocido")  
    Fin_Según_sea  
Fin_Metodo
```

Python

```
def convertir_a_romano():  
  
    a = int(input("Ingrese un número del 1 al 10:"))  
  
    if (a == 1):  
        romano = "I"  
    elif (a==2):  
        romano = "II"  
    elif (a==3):  
        romano = "III"  
    ...  
    else:  
        romano = "desconocido"  
  
    print(f"{a} en número romanos es {romano}")
```



EJERCICIO: Desarrolle un programa que implemente las operaciones básicas de una calculadora. La operación a realizar debe ser seleccionada de un menú en pantalla.

Recuerde que ya tenemos la clase diseñada. El menú de operación es debe ser como el siguiente:

1. Sumar
2. Restar
3. Multiplicar
4. Dividir

Calculadora

```
+ Real b
+ Real a

+ vacío pedir_datos()
+ Real sumar()
+ Real restar()
+ Real multiplicar()
+ Real dividir()
```

ESTRUCTURAS DE CONTROL SELECTIVAS



UNIVERSIDAD
DE ANTIOQUIA

Algoritmo

Clase AppCalculadora

```
publico vacio iniciar()
```

```
Calculadora cal = nueva Calculadora()
```

```
Entero opc
```

```
Real resul
```

```
Logico hizoLaOperacion = true
```

```
Escribir("Seleccione la operación a realizar: ")
```

```
Escribir("1. Sumar")
```

```
Escribir("2. Restar")
```

```
Escribir("3. Multiplicar")
```

```
Escribir("4. Dividir ")
```

```
Leer(opc)
```

```
cal.pedirDatos()
```

```
Según_sea (opc) haga
```

```
caso 1:
```

```
resul = cal.sumar()
```

```
Interrumpir
```

```
[...]
```

```
caso 4:
```

```
Si (cal.n2 != 0) entonces
```

```
resul = cal.dividir()
```

```
sino
```

```
Escribir("Error de división por 0")
```

```
hizoLaOperacion = false
```

```
Fin si
```

```
Interrumpir
```

```
En_otro_caso
```

```
Escribir("Seleccionó una opción incorrecta")
```

```
hizoLaOperacion = false
```

```
Fin Según_Sea
```

```
Si (hizoLaOperacion == true) entonces
```

```
Escribir("El resultado de la operación es: ", resul)
```

```
Fin Si
```

```
Fin Metodo
```

```
Fin_clase
```



EJERCICIO: Escriba una clase principal que permita a un conductor determinar el valor que debe pagar en un peaje, dependiendo del tipo de vehículo que este tiene. Considere las siguientes categorías de autos y los precios correspondientes:

- A. Autos: 10200
- B. Camioneta: 11800
- C. Camión: 18700
- D. Bus: 17400
- E. Otros: 23000



Algoritmo

Clase ValorPeaje

```
publico vacio calcular_valor_peaje()
```

```
Entero peaje
```

```
Cadena opc
```

```
Escribir("Seleccione el tipo de su vehículo: ")
```

```
Escribir("A. Automovil")
```

```
Escribir("B. Camioneta")
```

```
Escribir("C. Camión")
```

```
Escribir("D. Bus ")
```

```
Escribir("E. Otro")
```

```
Leer(opc);
```

```
Según_sea (opc) haga
```

```
caso "A":
```

```
    peaje = 10200
```

```
caso "B":
```

```
    peaje = 11800
```

```
caso "C":
```

```
    peaje = 18700
```

```
caso "D":
```

```
    peaje = 17400
```

```
en_otro_caso
```

```
    Escribir("No se sabe el tipo de auto")
```

```
    peaje = 18700
```

```
Fin_según_sea
```

```
Escribir("El valor del peaje es de: ", peaje)
```

```
Fin_metodo
```

```
Fin_clase
```

Python

```
peaje = 0

opc = input("Seleccione el tipo de vehículo: A. Automovil \nB. Camioneta \nC. Camión \nD. Bus \nE. Otro")

if (opc == "A"):
    peaje = 10200
elif (opc == "B"):
    peaje = 11800
elif (opc == "C"):
    peaje = 18700
elif (opc == "D"):
    peaje = 17400
else:
    print("No se sabe el tipo de auto")
    peaje = 18700

print(f"El valor del peaje es de: {peaje}")
```

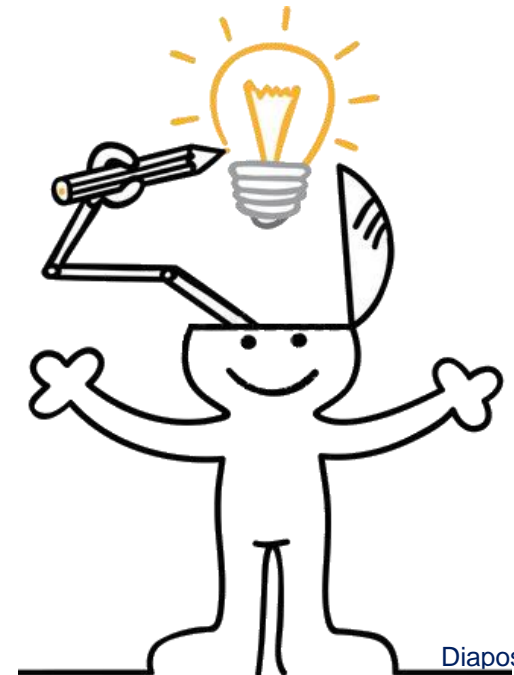
ESTRUCTURAS DE CONTROL SELECTIVAS



UNIVERSIDAD
DE ANTIOQUIA



EJERCICIO: Haga un algoritmo que dado el número del mes y el año indique cuantos días tiene dicho mes





**UNIVERSIDAD
DE ANTIOQUIA**

LÓGICA Y REPRESENTACIÓN I

PROGRAMA DE INGENIERÍA DE SISTEMAS

Material desarrollado por **Carlos Andrés Mera Banguero**

<https://github.com/carlosmera20/>