

Comprende los conceptos relacionados al control de estructuras condicionales simples en Python, a fin de emplearlos en el desarrollo de programas.

1. Introducción:

En esta parte se enseña al estudiante como debe diseñar condicionales basadas en las necesidades del ejercicio que simulan a las políticas o restricciones en un proceso de una organización; la forma y la ubicación de la condicional determinan la calidad de la información que se genera.

MÓDULO 3: Valores booleanos, Instrucciones if-elif-else, Bucles while y for, Control de flujo, Operaciones lógicas y bit a bit, Listas y arreglos.

2. Tomando decisiones en Python.

a. Las condiciones: preguntas y respuestas

Cuando desarrollamos un algoritmo, no siempre el desarrollo es con un trabajo secuencial que implique solo captura de datos, realizar ecuaciones de diversos tipos y mostrar un resultado, en algún momento puede generarse la necesidad de preguntar por algo.



Es en ese momento en el que debemos generar preguntas en nuestro algoritmo, de modo que podamos continuar con el proceso, iniciarlo o finalizarlo; lo cierto es que algo es claro respecto a las preguntas; nuestros computadores nunca aceptaran una pregunta ambigua, la pregunta debe ser claro que solo acepte uno de dos valores: TRUE o FALSE

b. Operadores de comparación

Los operadores que se usan para generar estas preguntas dentro de los algoritmos se denominan operadores de comparación y son similares a los de matemáticas (en concepto siguen siendo los mismos, sin embargo, en simbología puede modificarse dependiendo el lenguaje de programación).

SÍMBOLO	DESCRIPCION	EJEMPLO	VALOR BOOLEANO
==	Igual que	6==4	False
!=	Diferente que	Perú!=Chile	True
<	Menor que	Carlos < Carla	False
>	Mayor que	45>23	True
<=	Menor o igual que	2<=10	True
>=	Mayor o igual que	Victor>=Juan	True

Si observamos la tabla, podremos ver claramente los símbolos que representan en Python a los operadores de comparación o relacionales.

Por otro lado, puede ver algo en los ejemplos, que le podría interesar a estas alturas, y es la pregunta (Perú!=Chile) o (Carlos < Carla), estas preguntas tienen un valor de verdad asociado, la pregunta es ¿cómo se llega a ese resultado?

```
print("Perú"!="Chile")  
print("Carlos"<"Carla")
```

Al ejecutar, el resultado es el siguiente:

```
True  
False
```

Si hablamos de números, identificar ello es fácil, porque conocemos el valor de estos y son de aplicación numérica; pero cuando tenemos textos ello no ocurre; entonces, ¿cómo es que la máquina puede devolver un valor de verdad respecto a esas preguntas?, la respuesta se encuentra en una de las tablas más antiguas que se usaron para codificar cada carácter y poder hacer que un computador pueda comprender que operaciones deseamos que realice, esta es la tabla de códigos ASCII.

32		58	:	83	S	108	l	133	à	158	×	183	À	208	ø	233	Ú
33	!	59	;	84	T	109	m	134	á	159	÷	184	©	209	ð	234	Û
34	"	60	<	85	U	110	n	135	â	160	×	185	ª	210	É	235	Ü
35	#	61	=	86	V	111	o	136	ä	161	í	186	»	211	Ê	236	Ý
36	\$	62	>	87	W	112	p	137	å	162	ó	187	¼	212	Ë	237	Ÿ
37	%	63	?	88	X	113	q	138	æ	163	ô	188	½	213	Ì	238	˘
38	&	64	@	89	Y	114	r	139	ç	164	ñ	189	¾	214	Í	239	˙
39	'	65	A	90	Z	115	s	140	ê	165	Ň	190	¥	215	Î	240	˚
40	(66	B	91	[116	t	141	ë	166	°	191	₹	216	Ï	241	±
41)	67	C	92	\	117	u	142	Ä	167	²	192	₠	217	Ĵ	242	ˆ
42	*	68	D	93]	118	v	143	Å	168	³	193	₡	218	Ķ	243	‰
43	+	69	E	94	^	119	w	144	Ä	169	©	194	₢	219	Ĺ	244	¶
44	,	70	F	95	_	120	x	145	Æ	170	¬	195	₣	220	Ł	245	§
45	-	71	G	96	`	121	y	146	Œ	171	½	196	₤	221	Ű	246	+
46	.	72	H	97	a	122	z	147	Œ	172	¼	197	₥	222	Ų	247	ˆ
47	/	73	I	98	b	123	{	148	Š	173	¿	198	₦	223	Ŵ	248	˚
48	0	74	J	99	c	124		149	Š	174	«	199	₧	224	Ŷ	249	˘
49	1	75	K	100	d	125	}	150	Ź	175	»	200	₨	225	Ÿ	250	˙
50	2	76	L	101	e	126	~	151	ž	176	ˆ	201	₪	226	Ž	251	ˆ
51	3	77	M	102	f	127	¸	152	ÿ	177	˜	202	€	227	Ų	252	ˆ
52	4	78	N	103	g	128	¸	153	Ų	178	˜	203	₭	228	Ŷ	253	ˆ
53	5	79	O	104	h	129	ù	154	Ų	179	˜	204	₮	229	Ų	254	ˆ
54	6	80	P	105	i	130	é	155	ø	180	˜	205	₯	230	µ	255	ˆ
55	7	81	Q	106	j	131	â	156	£	181	À	206	₰	231	þ		
57	9	82	R	107	k	132	ä	157	Ø	182	Ä	207	₱	232	þ		

Cuando una computadora analiza caracteres en general lo hace uno por uno de izquierda a derecha, mientras que NO exista igualdad siempre buscará que sigue hacia la derecha, cuando no haya nada o encuentre desigualdad emitirá un valor de verdad.

Dice: Perú!=Chile

P= código 80

C= código 67

Entonces 80!=67 -----> **True**

Finalmente, si deseamos ver a que equivale ese código 80 o 67, se debe transformar a base 2, de esa manera finalmente la computadora tendrá su equivalente en bits para que comprenda de que se trata.

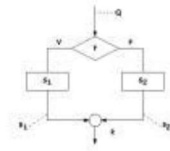
c. Las condiciones y su ejecución.



Hasta el momento tenemos la idea de que ocurre cuando se genera una pregunta y cómo es que el computador interpreta la pregunta, el problema se centra ahora en cómo hacer ello posible dentro de un algoritmo, para ello Python nos define a la Instrucción IF.

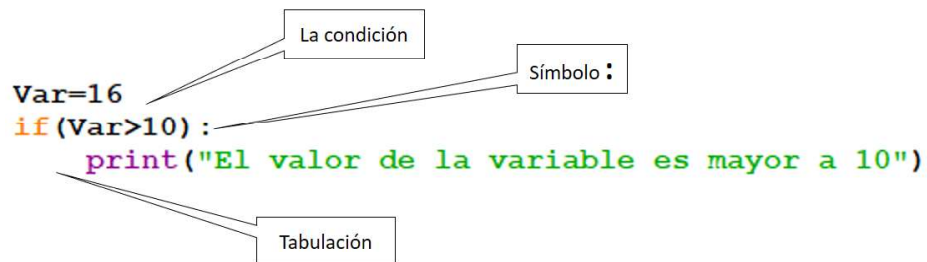
d. Sentencia o declaración if.

Vamos a estudiar entonces a los algoritmos que hacen uso de estructuras condicionales, este tipo de algoritmos se caracteriza por que entre sus instrucciones no solo se muestran estructuras secuenciales comunes sino también estructuras condicionales, pero aún no se muestran estructuras repetitivas.



- **Sentencia if**

Esta Sentencia en su estructura simple contiene algunas partes bien definidas:



- Al lado derecho del if, se ubica la condición que se analizará (esta puede ser una variable o un valor lógico).
- Luego de la condición se ubica el símbolo dos puntos (:), el cual indica que inicia una secuencia de instrucciones.
- Podemos observar que en la siguiente instrucción de la sentencia if, esta se encuentra tabulada (identada, es otro término a la tabulación), esto le indica a Python que la o las instrucciones siguientes pertenecen a la sentencia **if**.

- **La condición**

Cuando diseñamos una condición, debemos tener en cuenta la siguiente estructura:

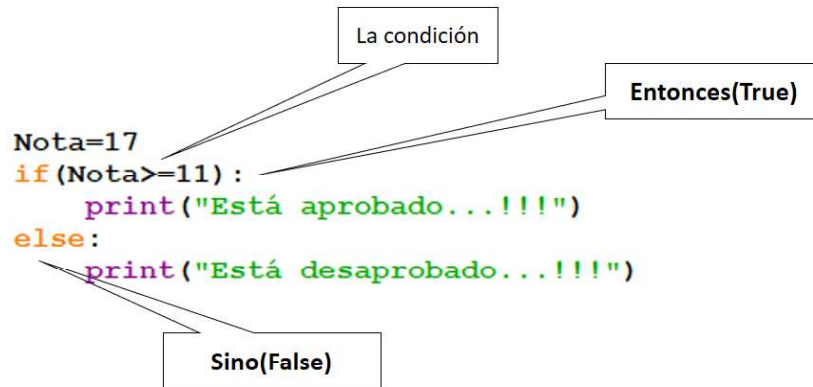
Expresión1 O.R. Expresión2

- Donde las Expresiones 1 o 2 pueden ser variables, constantes, ecuaciones entre otros.
- La O.R. se refiere a los operadores relacionales.
- Es ideal que las condiciones estén descritas entre paréntesis por un tema de orden.
- Cuando se evalúa una condición en la instrucción if, se debe considerar que finalmente lo que requiere if es solo un valor (True o False).

False) para continuar con su desarrollo, por lo que puede evaluar un valor lógico sin una estructura como la indicada, tener en cuenta ello.

e. Declaración if – else.

Observemos a la sentencia if, implementada en su totalidad:

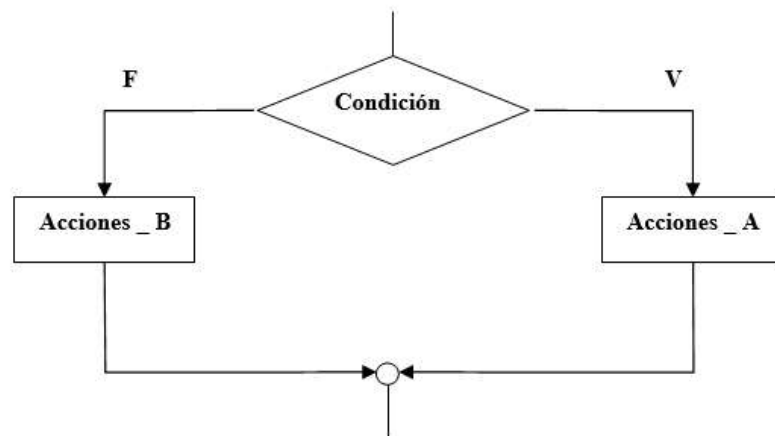


- La condición al ser evaluada solo podrá ser verdadera o falsa, en el gráfico se muestra la ubicación de las sentencias o instrucciones que acompañan a cada una de ellas.
- La zona ubicada después de los dos puntos, indica las instrucciones del lado verdadero de la condición.
- La zona ubicada después de la instrucción **else** y los dos puntos, indica las instrucciones del lado falso de la condición.
- Ambos lados tienen tabulación para ser identificadas como instrucciones de cada lado.

3. Graficando la sentencia if.

a. Diagrama de flujo: estructura y simbología.

El diagrama que se asocia a esta sentencia, implica el uso del rombo como símbolo principal y el uso de conectores para indicar la secuencia de inicio y fin de la estructura.



El pseudocódigo asociado a la estructura anterior es el siguiente:

```
Si (Condición) entonces:  
    Realizar acciones _ A  
Sino  
    Realizar acciones _ B  
Fin si
```

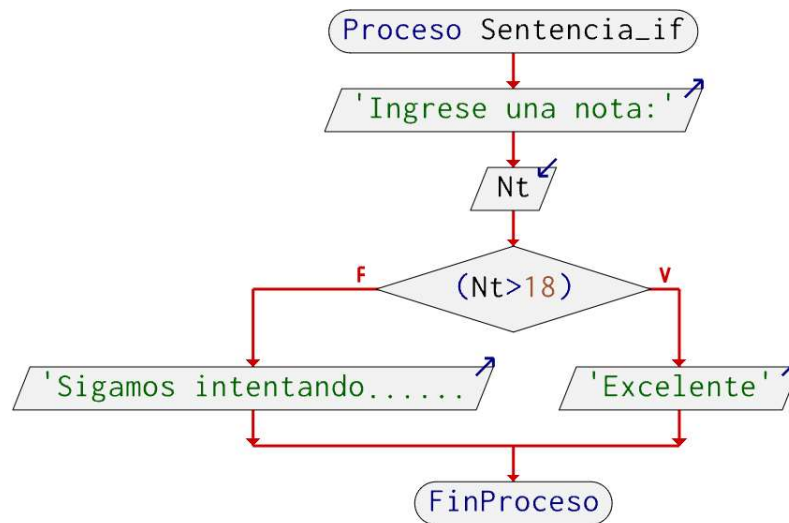
Veamos un ejemplo:

La idea es ingresar una nota y si esta sobrepasa el valor de 18 colocar "Excelente" como mensaje caso contrario colocar "Sigamos intentando"

Pseudocódigo:

```
1  Proceso Sentencia_if  
2      Escribir "Ingrese una nota:"  
3      leer Nt  
4      Si (Nt>18) Entonces  
5          Escribir "Excelente"  
6      SiNo  
7          Escribir "Sigamos intentando...!!!"  
8      Fin Si  
9  FinProceso  
10
```

Diagrama de Flujo:



Observación: algunos autores indican que el lado verdadero de la sentencia en el gráfico debe estar ubicado por el lado izquierdo, otros incluso colocan el lado verdadero por el lado inferior del rombo; lo cierto es que al final la idea es que la condición de la sentencia IF solo podrá ser verdadera o falsa, el resto solo implica adaptarse a símbolos y diagramas.

b. Desarrollo de ejercicios tipo

Los ejercicios que veremos a continuación implican el conocer los realizado en estructuras secuenciales, ya que mucho de lo que veremos a continuación busca implementar condiciones a los ejercicios secuenciales; en este bloque de ejercicios ya no consideraremos el pseudocódigo, pero si el diagrama de flujo.

Condicionales Simples

1. Problema Prg1

Diseñar un algoritmo que permita ingresar un número, y luego mediante un mensaje indique si fue positivo.

Solución:

a. Análisis.

¿Qué te piden que realices?

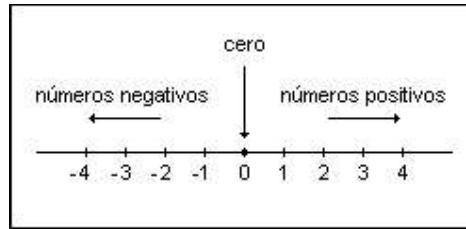
Evaluar un número, si fue positivo

¿Qué datos necesito conocer?

El número.

b. Planteamiento Lógico.

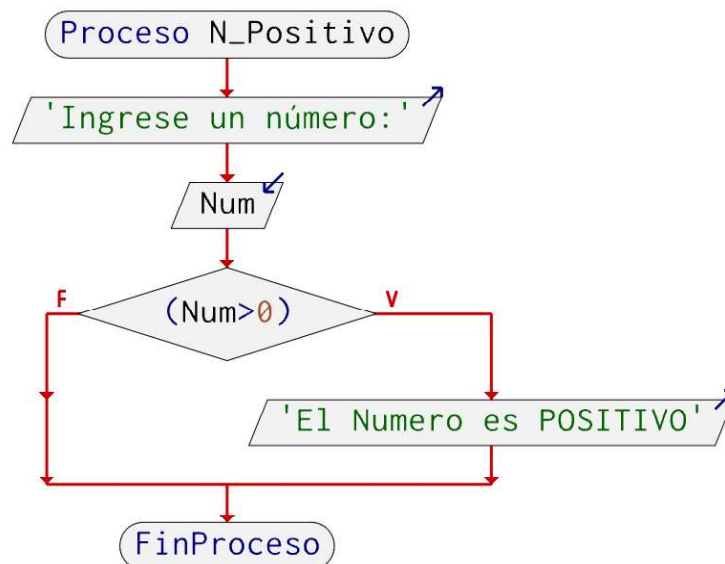
Para poder saber si un número es positivo, debemos recordar la recta numérica y luego analizar cuál sería la condicional que se requiere aplicar:



Si observamos el gráfico, podemos concluir que todos los números del cero a la derecha son positivos, del cero a la izquierda son negativos y el cero es neutro, es decir no tiene signo; por lo tanto, para saber cómo preguntar por un número positivo debemos considerar: **Un Número >0**

c. Diseño.

Diagrama de Flujo:



En el diagrama mostrado podemos observar que no usamos el lado FALSO, ello es porque en el ejercicio no nos indicaron que ocurriría por dicho lado, así que simplemente se dejó sin usar.

No necesariamente usted debe usar toda la estructura, procure primero cubrir todo lo solicitado, luego haga las mejoras o adecuaciones que considere necesarias, sin salir de la lógica en general.

d. Lenguaje de Programación.

```
print("Ingrese un número:")  
Num=int(input())  
if (Num>0):  
    print("El Número es POSITIVO")
```

Al ejecutar el programa, se muestra lo siguiente:

```
Ingrese un número:  
5  
El Número es POSITIVO
```

Observación: Si no se ingresa un número positivo, el resultado del programa será **nada**, ya que no se contempla que hacer por el lado falso.

2. Problema Prg2

Diseñar un algoritmo que permita ingresar un número, y luego indicar mediante un mensaje si fue par o impar.

Solución:

a. Análisis.

¿Qué te piden que realices?

Evaluar un número si fue par o impar.

¿Qué datos necesito conocer?

El número.

b. Planteamiento Lógico.

Cuando analizamos los números enteros en general, estos pueden ser pares o impares, un concepto asociado a la divisibilidad y para el caso sería divisibilidad por 2 o múltiplos de 2.

Por otro lado, cabe entonces recordar porque decimos divisible o múltiplo de un número, la respuesta es simple, cuando se divide un número por un valor y esta división resulta ser exacta, se define que le número evaluado es como múltiplo de dicho valor, por ejemplo:

$$\begin{array}{r|l} 18 & 2 \\ \hline 18 & 9 \\ \hline 0 \end{array}$$

$$\begin{array}{r|l} 31 & 2 \\ \hline 30 & 15 \\ \hline 1 \end{array}$$

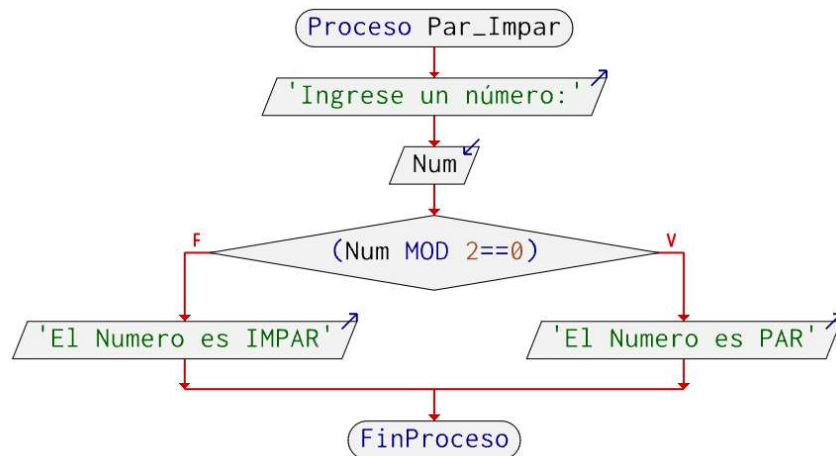
La primera división es exacta, por lo que podemos decir que 18 es múltiplo de 2, mientras que en la segunda división ello no ocurre.

En el caso de Python, debemos aplicar el operador %, para poder obtener el resto de la división y tal como observamos, toda vez que el resto entregue cero de resultado en una operación, podemos decir que es divisor o múltiplo; para el caso de los pares, la ecuación será de este modo entonces: **Numero%2==0**

c. Diseño.

Diagrama de Flujo:

Considerar que, en el PSeint el operador % es reemplazado por el operador MOD, que cumple la misma función.



d. Lenguaje de Programación.

```
print("Ingrese un número:")
Num=int(input())
if (Num%2==0):
    print("El número es PAR")
else:
    print("El número es IMPAR")
```

Al ejecutar el programa, se muestra lo siguiente:

```
Ingrese un número:
17
El número es IMPAR
```

3. Problema Prg3

Diseñar un algoritmo que permita ingresar tres notas, genere el promedio y luego indique mediante un mensaje si aprobó o desaprobó el curso, considerar que en el IESTP se aprueba con 14 y se tiene medio punto a favor.

Solución:

a. Análisis.

¿Qué te piden que realices?

Indicar si un alumno aprobó o desaprobó un curso

¿Qué datos necesito conocer?

Las tres notas.

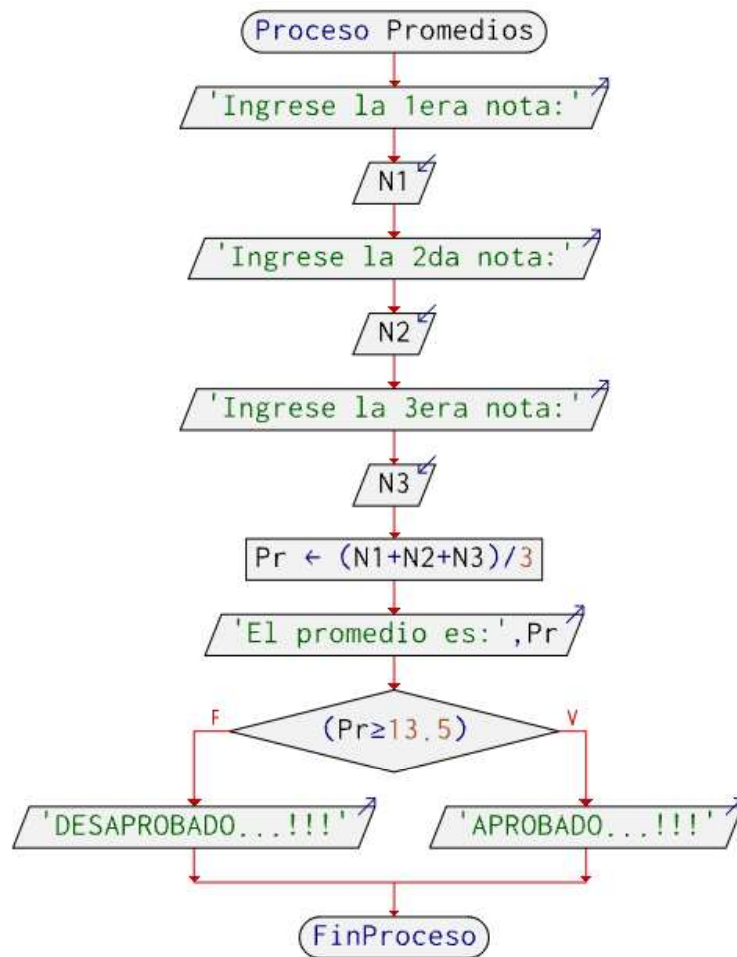
b. Planteamiento Lógico.

En este caso, lo primero a indicar es que se debe calcular el promedio, pero ello ya se ha tratado anteriormente cómo se desarrolla; ahora solo se tiene que generar la condición adecuada para poder indicar si un alumno aprueba o no un curso.

El enunciado indica que se aprueba con 14 y se considera medio punto a favor del alumno (ello se aplica en el redondeo), entonces podemos decir que aprueban con 13.5; a lo que podemos concluir que la condición sería de la siguiente forma: **Promedio \geq 13.5**

c. Diseño.

Diagrama de Flujo:



Debe considerarse que en PSeint el símbolo relacional para indicar Mayor o igual que, se traduce en el estilo de las matemáticas: \geq

d. Lenguaje de Programación.

```

print("Ingrese 1era Nota:")
N1=float(input())
print("Ingrese 2da Nota:")
N2=float(input())
print("Ingrese 3era Nota:")
N3=float(input())
Pr=(N1+N2+N3)/3
print("El promedio es:",Pr)
if (Pr>=13.5):
    print("APROBADO...!!!")
else:
    print("DESAPROBADO...!!!")
  
```

Considerar que seguimos asumiendo que el usuario ingrese valores de notas correctas.

Al ejecutar el programa, se muestra lo siguiente:

```
Ingrese 1era Nota:
14
Ingrese 2da Nota:
18
Ingrese 3era Nota:
11
El promedio es: 14.333333333333334
APROBADO...!!!
```

4. Problema Prg4

Ingresar un número de dos cifras, mediante un algoritmo indique con un mensaje si sus componentes fueron iguales, ejemplo: 44 tiene componentes iguales.

Solución:

a. Análisis.

¿Qué te piden que realices?

Indicar si los componentes de un número de 2 cifras son iguales

¿Qué datos necesito conocer?

El número de 2 cifras

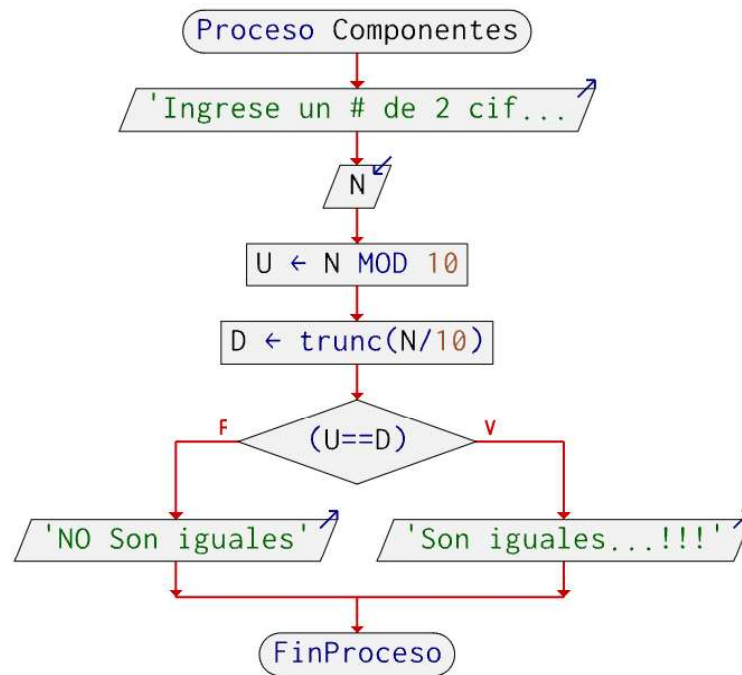
b. Planteamiento Lógico.

Anteriormente ya hemos visto cómo se descompone un número, y es obvio que es lo primero que debemos desarrollar en esta ocasión, para luego preguntar por sus componentes, por decir:

Unidades==Decenas

c. Diseño.

Diagrama de Flujo:



d. Lenguaje de Programación.

```

print("Ingrese un # de 2 cifras:")
N=int(input())
U=N%10
D=N//10
if (U==D):
    print("Son iguales...!!!")
else:
    print("NO Son iguales...!!!")
  
```

Al ejecutar el programa, se muestra lo siguiente:

```

Ingrese un # de 2 cifras:
56
NO Son iguales...!!!
  
```

5. Problema Prg5

Diseñar un algoritmo que permita ingresar 3 números, luego evaluarlos e indicar la suma de todos los que fueron menores a 10 (No puede usar acumuladores).

Solución:

a. Análisis.

¿Qué te piden que realices?

La suma de los números menores a 10

¿Qué datos necesito conocer?

Los 3 números

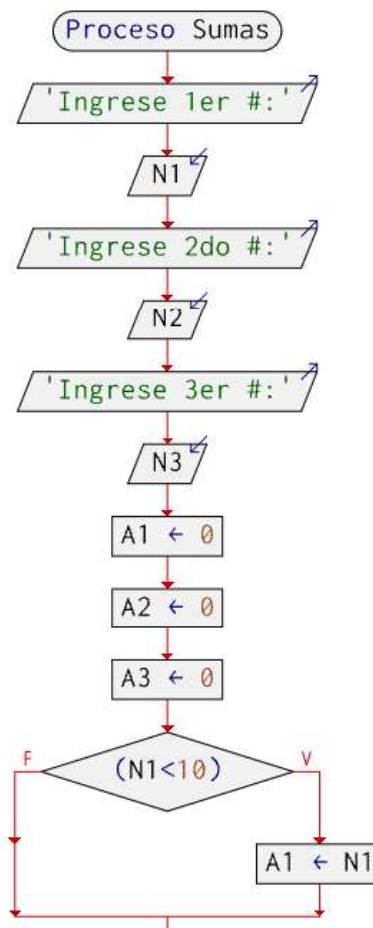
b. Planteamiento Lógico.

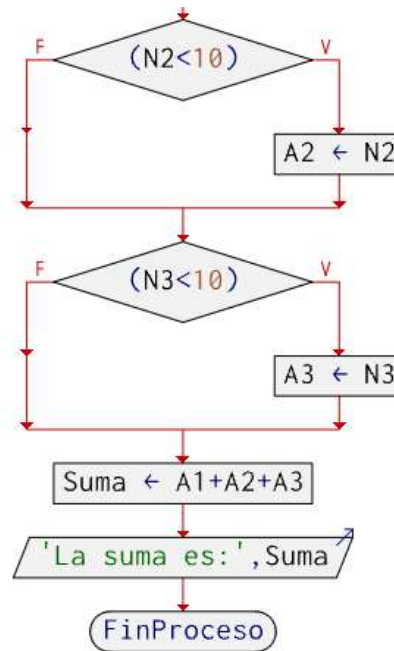
Este problema plantea una solución que no hemos visto al momento y es el uso de varias condiciones en un ejercicio, eso sí, estas no estarán anidadas, sino que deberán generar una secuencia de preguntas independientes, ya que solicitan analizar 3 números por separado e incluso no podemos usar acumuladores para sumarlos si son menores a 10.

Usaremos variables entonces para lograr esa suma.

c. Diseño.

Diagrama de Flujo:





d. Lenguaje de Programación.

```

print("Ingrese 1er #:")
N1=int(input())
print("Ingrese 2do #:")
N2=int(input())
print("Ingrese 3er #:")
N3=int(input())
A1=0;A2=0;A3=0
if (N1<10):
    A1=N1
if (N2<10):
    A2=N2
if (N3<10):
    A3=N3
Suma=A1+A2+A3
print("La suma es:",Suma)
  
```

Observar cómo se inicializan variables consecutivas

Al ejecutar el programa, se muestra lo siguiente:

```

Ingrese 1er #:
18
Ingrese 2do #:
5
Ingrese 3er #:
6
La suma es: 11
  
```

6. Problema Prg6

Diseñe un algoritmo que permita ingresar el estatus de un colaborador de una empresa (activo o inactivo) y su sueldo, en caso de que esté activo deberá adicionarle el 20% a su sueldo; caso contrario indicar que el sueldo equivale a cero; mostrar el sueldo final, en cualquier caso.

Solución:

a. Análisis.

¿Qué te piden que realices?

Mostrar el sueldo final

¿Qué datos necesito conocer?

El estado y el sueldo del colaborador

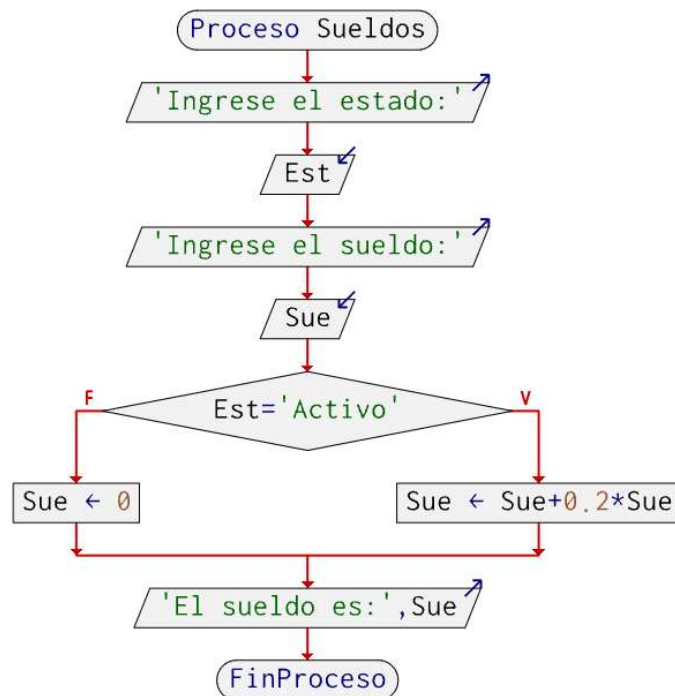
b. Planteamiento Lógico.

En este caso se debe aplicar el porcentaje respectivo de acuerdo al estado del colaborador que es un valor textual, así que debemos tener cuidado al momento de su ingreso y operación, ya que el estado tiene un tratamiento especial.

Estos problemas son de comprensión lectora, así que tener leer con cuidado y hacerle el seguimiento respectivo.

c. Diseño.

Diagrama de Flujo:



d. Lenguaje de Programación.

```

print("Ingrese el estado:")
Est=input()
print("Ingrese el sueldo:")
Sue=float(input())
if(Est=="Activo") :
    Sue=Sue+0.2*Sue
else:
    Sue=0
print("El sueldo es:",Sue)

```

Observe la condición
ejercida hacia un dato
de texto

Se asigna cero,
porque no
tiene sueldo

Al ejecutar el programa, se muestra lo siguiente:

```

Ingrese el estado:
Activo
Ingrese el sueldo:
1000
El sueldo es: 1200.0

```

Observe esta variante al código, podrá ver lo siguiente al momento de ejecutarlo:

- Una lista de opciones (tipo menú) que deberá digitar en el estado.
- Se está calculando el adicional en una variable a parte, para después ser aplicada al sueldo.
- Finalmente, podrá observar que en el print() final está la instrucción f, que es usada para dar formatos de salida.

```

print("Ingrese el estado:\n"
      "Activo\n"
      "Inactivo")
Est=input()
print("Ingrese el sueldo:")
Sue=float(input())
if(Est=="Activo") :
    Adic=0.2*Sue
else:
    Adic=0
Sue=Sue+Adic
print(f"El sueldo es:{Sue}")

```

7. Problema Prg7

En una tienda se compran diversos productos, en ese sentido al momento de adquirir un producto siempre se indica el precio y la cantidad a comprar; de ese modo se obtiene: $\text{Subtotal} = \text{Precio} * \text{Cantidad}$

A toda compra se aplica un descuento del 26% siempre en cuando el subtotal obtenido sea mayor a 100 soles. Por otro lado, existe un impuesto

que se debe obtener una vez realizado el descuento que equivale al 5% y que lo pagarán todas las compras; Finalmente, una vez aplicado todos los descuentos e impuestos, si el subtotal resulta siendo mayor a 5000 soles se paga un impuesto adicional del 8%.

Mediante un algoritmo determine cuanto pagará la persona por una compra.

Solución:

a. Análisis.

¿Qué te piden que realices?

El pago a realizar por la venta de un producto

¿Qué datos necesito conocer?

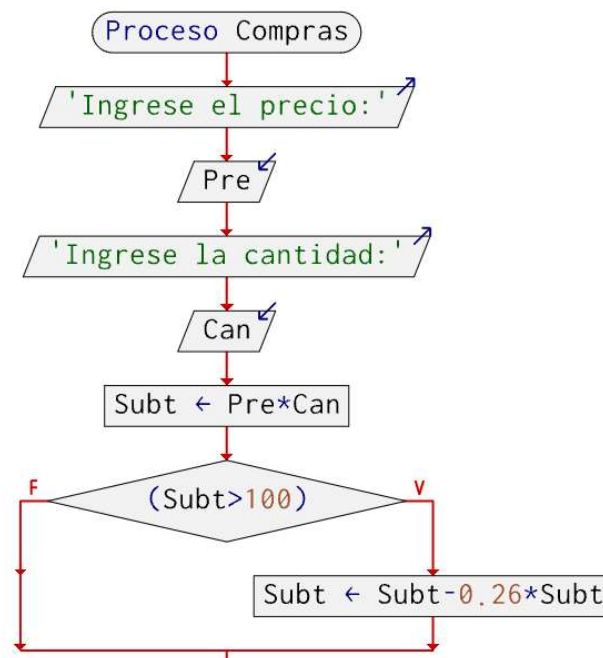
El precio y la cantidad

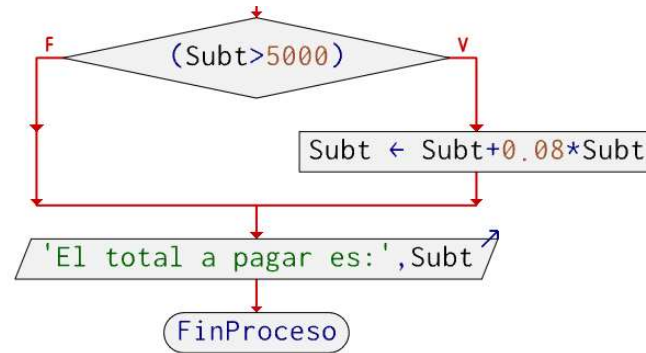
b. Planteamiento Lógico.

Este problema es de seguimiento, requiere de varios cálculos iniciando por hallar el subtotal, para luego hallar los descuentos e impuestos; que finalmente determinarán el total a pagar. Tener mucho cuidado con la comprensión lectora y el orden de los cálculos.

c. Diseño.

Diagrama de Flujo:





d. Lenguaje de Programación.

```

print("Ingrese el precio:")
Pre=float(input())
print("Ingrese la cantidad:")
Can=float(input())
Subt=Pre*Can
if (Subt>100):
    Subt=Subt-0.26*Subt
Subt=Subt+0.05*Subt
if (Subt>5000):
    Subt=Subt+0.08*Subt
print("El total a pagar es:", Subt)

```

Al ejecutar el programa, se muestra lo siguiente:

```

Ingrese el precio:
10
Ingrese la cantidad:
30
El total a pagar es: 233.1

```

Observe esta variante al código, observar las consideraciones que se aplican a este diseño:

- Se considera asignar a cada operación una variable de modo que se puede detallar cada una, esto puede ser necesario cuando los resultados de cada cálculo se requieren reutilizar más adelante en el código.
- Debe observar cómo se controla cada variable dentro de la condicional, ya que, si no cumple con la condición, se le asigna el valor de cero.
- Se muestra los resultados de cada cálculo realizado, mediante la función `print()`.

```
print("Ingrese el precio:")
Pre=float(input())
print("Ingrese la cantidad:")
Can=float(input())
Subt=Pre*Can
print("Subtotal:", Subt)
if (Subt>100):
    Dsc=0.26*Subt
else:
    Dsc=0
Subt=Subt-Dsc
Imp1=0.05*Subt
Subt=Subt+Imp1
if (Subt>5000):
    Imp2=0.08*Subt
else:
    Imp2=0
Total=Subt+Imp2
print("Descuento aplicado:", Dsc)
print("Impuesto 1 aplicado:", Imp1)
print("Impuesto 2 aplicado:", Imp2)
print("El total a pagar es:", Total)
```

La ejecución con los mismos valores que el ejercicio anterior se verán de esta forma:

```
Ingrese el precio:
10
Ingrese la cantidad:
30
Subtotal: 300.0
Descuento aplicado: 78.0
Impuesto 1 aplicado: 11.100000000000001
Impuesto 2 aplicado: 0
El total a pagar es: 233.1
```

Puede observar el detalle de lo calculado en cada línea mostrada, esto también le puede servir cuando desea mostrar o saber que valores están pasando por las variables, muchas veces se desarrollan cálculos que asumimos están bien, sin embargo, puede que contengan errores, una de las formas más simples de control es el uso de mensajería (print()).

4. Actividad

A) Ejercicios en General

Desarrolle los siguientes ejercicios, considerando lo desarrollado anteriormente:

- a) Desarrolle un algoritmo que permita ingresar un número por teclado, luego a este deberá sumarle 10, seguidamente se deberá observar si el número que se genera es mayor a 50 si ello ocurre sumarle 100 en caso de no ser así deberá restarle 80; finalmente después de todo lo anterior, muestre el número generado.
- b) Se ingresan 3 números, indicar si el 3er número es igual a la suma de los dos primeros números ingresados; use un mensaje.
- c) Realice un algoritmo que permita ingresar dos números de tres cifras, sumarlos y muestre con un mensaje si resultado obtenido fue par o impar.
- d) Diseñar un algoritmo que permita calcular y mostrar el Total_pago por una persona que compra un artículo, si se sabe que: si el monto a pagar finalmente sobrepasa los 100 soles se le aplica un descuento del 20%, en caso contrario solamente se le aplica un descuento del 5%.

$$\begin{aligned}\text{Importe a Pagar} &= \text{Precio} * \text{Cantidad} \\ \text{Total_pago} &= \text{Importe} - \text{Descuento}\end{aligned}$$

- e) Se van a ingresar 3 notas, determinar el promedio de estos, si se sabe que este se obtiene quitando la menor nota, adicionalmente mostrar un mensaje que diga si están o no aprobados.

Ejemplo:

Sean las notas: 19, 16 y 13

Se deberá promediar solamente 19 y 16 (la menor no se toma en cuenta)

OBS: No usar if anidados

- f) Un obrero necesita calcular su salario semanal, el cual se obtiene de la siguiente manera: Si trabaja 40 horas o menos en una semana se le paga \$16 por hora; Si trabaja más de 40 horas se le paga \$18 por cada hora extra; Desarrolle un algoritmo que permita mostrar cuál es el salario final del Obrero en una semana, además tener en cuenta que el empleador le paga un seguro semanal que equivale al 25% de los que gana.
- g) Se va ingresar un número de 3 cifras, indique si este es capicúa.

Ejemplos de números capicúas: 121 / 343 / 434.

- h) Se van ingresar 2 número, indique con un algoritmo cual fue el mayor.
- i) Desarrolle un algoritmo que permita hallar el valor de Y, si se tiene:

$$Y = 4X + \frac{X + 1}{X - 1}$$

Tenga en cuenta los valores de X para los que no se puede desarrollar la ecuación por generar división entre cero.

- j) Desarrolle un algoritmo que permita hallar la Velocidad Final, si se sabe:

$$Vm = \frac{Vo + Vf}{2}$$

Finalmente indique si la velocidad obtenida es similar a la velocidad inicial, use un mensaje.

5. Fuentes consultadas:

- A) <https://edube.org/learn/programming-essentials-in-python-part-1-spanish>
- B) Edison Zavaleta C. (2005). *Fundamentos de Programación*. Perú, Editorial Abaco-Lima.