

Instructivo # 3

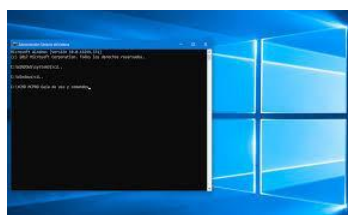
Programas Condicionales

Objetivo: Realizar programas en Python, donde se visualicen estructuras condicionales simples y múltiples

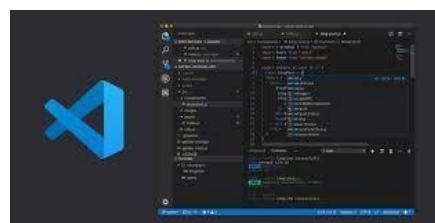
Herramienta de trabajo: Software



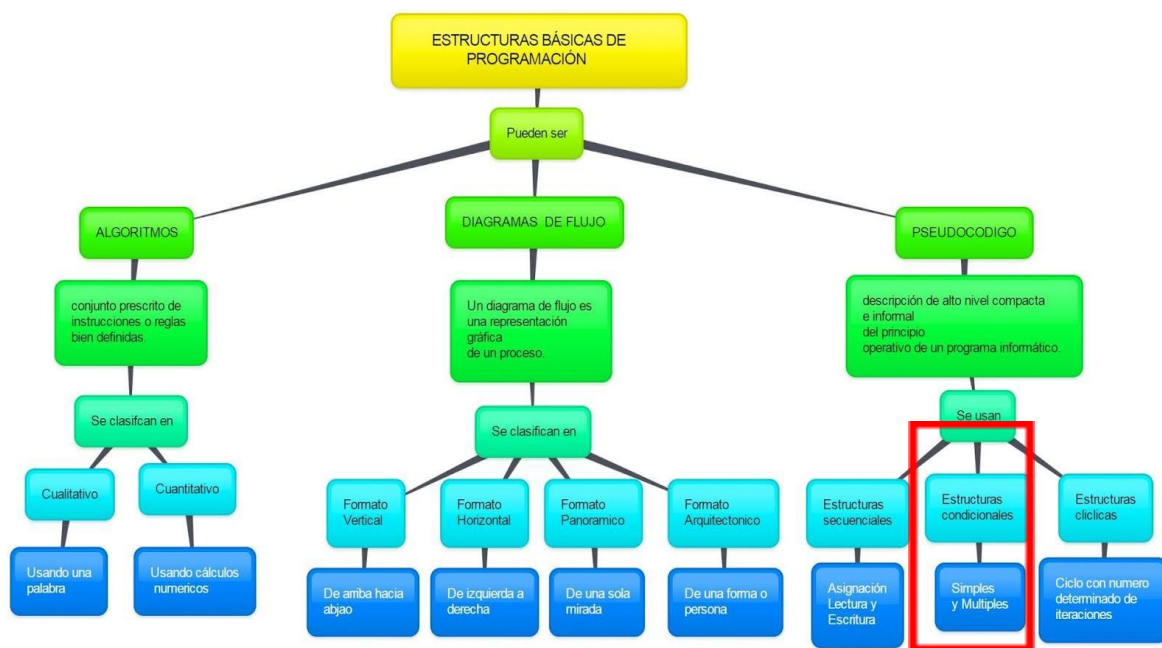
Python 3.9



MSDOS



Editor Visual Studio Code
(Terminal),



created with www.bubbl.us



Bloques de Código en Python

¿Qué es la **indentación**? En un lenguaje informático. La indentación es lo que la sangría al lenguaje humano escrito (a nivel formal). Así como para el lenguaje formal, cuando uno redacta una carta, debe respetar ciertas sangrías, los lenguajes informáticos, requieren una indentación.

No todos los lenguajes de programación, necesitan de una indentación, aunque sí, se estila implementarla, a fin de otorgar mayor legibilidad al código fuente. **Pero en el caso de Python, la indentación es obligatoria**, ya que, de ella, dependerá su estructura.

En la siguiente imagen, se observa el inicio, contenido y finalización de un bloque.





Una **indentación de 4 (cuatro) espacios en blanco**, indicará que las instrucciones indentadas, forman parte de una misma estructura de control. Se define de la siguiente forma.

```
inicio de la estructura de control:  
    expresiones
```

La **codificación (encoding)**, es otro de los elementos del lenguaje que no puede omitirse a la hora de hablar de estructuras de control. Es así, que se edita el siguiente parámetro al inicio de un archivo Python, que indica al sistema, la codificación de caracteres utilizada en el archivo.

```
# -*- coding: utf-8 -*-  
  
print ("En el Ñágara encontré un Ñandú")
```

Terminal:

```
En el Ñágara encontré un Ñandú
```

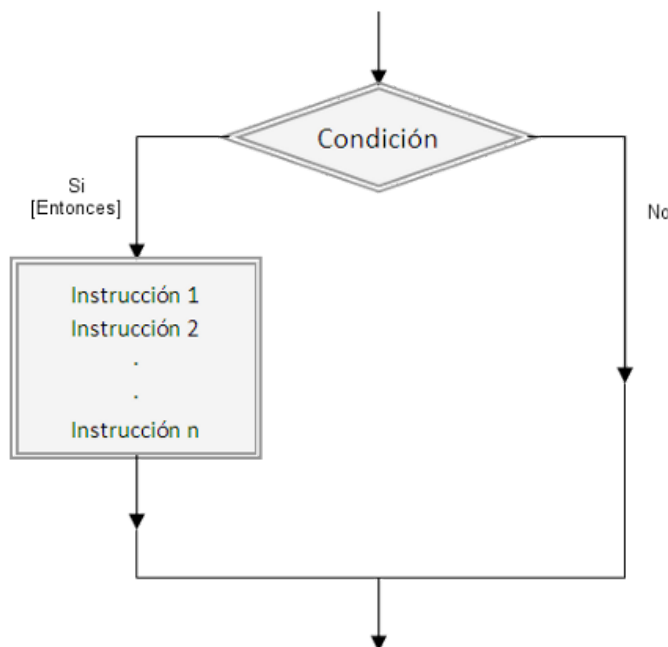
Estructura de control de decisión o selección

uso de **if** (si), **elif** (sino, si) y **else** (sino).

Las estructuras de control de flujo condicionales, se definen mediante el uso de tres palabras claves reservadas, del lenguaje: if (si), elif (sino, si) y else (sino).

Se utilizan para tomar decisiones lógicas, en el caso cuando existe un número de posibles alternativas resultantes de la evaluación de una determinada condición.

Selección Simple: **if – else**



if (condicion):

#hago algo

if (condicion):

#hago algo



else:

#hago otra cosa

Ejercicio 31.

Solicitar al usuario un **número**. Si el **número es el 1000**, imprimir "Ganaste un premio".

```
print("Juego de la Loteria")
numero=int(input("Señor usuario, digitar un numero del 1 al 5000:"))
if numero==1000:
    print("Ganaste un premio!")
```

Decisión:

```
if numero==1000:
    print("Ganaste un premio!")
```

Terminal (prueba de escritorio):

```
N. de cliente:3
PS C:\0_SENA_Armenia_2021\3_Guias_2021\1_Registros_Guias_2021\GUIAS_2021> python ejercicio31.py
Juego de la Loteria
Señor usuario, digitar un numero del 1 al 5000:1000
Ganaste un premio!
```



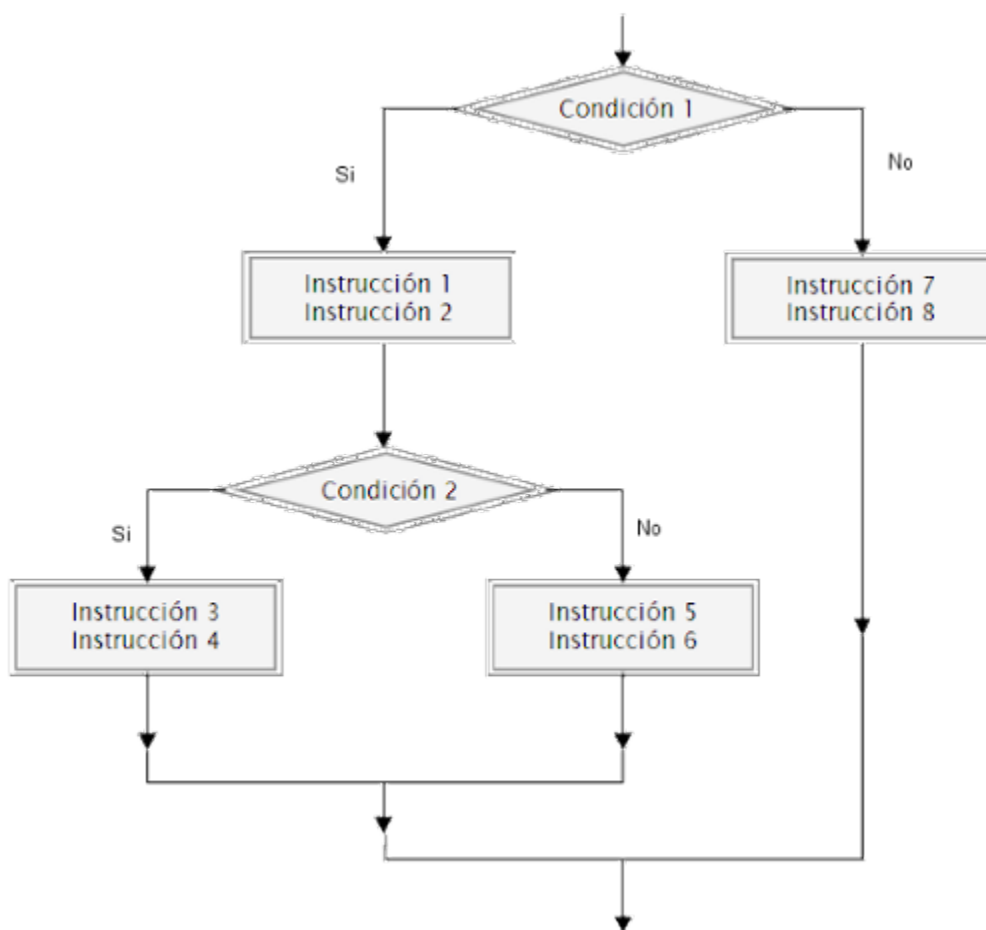
Ejercicio 32.

Solicitar al usuario que ingrese **dos números** y mostrar cuál de los dos es menor. No considerar el caso en que ambos números son iguales.

else:

```
print("Ejercicio 32: El menor de dos numeros")
numero1=int(input("Ingresar el primer numero:"))
numero2=int(input("Ingresar el segundo numero:"))
if numero1<numero2:
    print(f"el primero es menor:{numero1}")
else:
    print(f"el segundo es menor:{numero2}")
```

Selección Multiple: **if** (si), **elif** (sino, si) y **else** (sino).



if (condicion):

#hago algo

elif (condicion 2):

#hago algo otra cosa

else:

#por descarte

Ejercicio 33.

Solicitar al usuario que ingrese dos números y mostrar cuál de los dos es menor.
Considerar el caso en que ambos números son iguales.

elif:

```
numero1=int(input("Un número:"))
numero2=int(input("Otro número:"))
if numero1<numero2:
    print("El primero es menor")
elif numero2<numero1:
    print("El segundo es menor")
else:
    print("Son iguales")
```

Para describir la evaluación a realizar sobre una condición, se utilizan operadores relacionales (o de comparación):



Símbolo	Significado	Ejemplo	Resultado
==	Igual que	5 == 7	False
!=	Distinto que	rojo != verde	True
<	Menor que	8 < 12	True
>	Mayor que	12 > 7	True
<=	Menor o igual que	12 <= 12	True
>=	Mayor o igual que	4 >= 5	False

Y para evaluar más de una condición simultáneamente, se utilizan operadores lógicos:

Operador	Ejemplo	Explicación	Resultado
and	5 == 7 and 7 < 12	False and False	False
and	9 < 12 and 12 > 7	True and True	True
and	9 < 12 and 12 > 15	True and False	False
or	12 == 12 or 15 < 7	True or False	True
or	7 > 5 or 9 < 12	True or True	True
xor	4 == 4 xor 9 > 3	True o True	False
xor	4 == 4 xor 9 < 3	True o False	True



Ejercicio 34_1.

Se requiere realizar un programa donde se le indique al usuario cuales son los días de la semana de trabajo y días de descanso.

```
ejercicio34_1.py > ...
1  #Dias de la semana
2
3  dia=input("ingrese porfavor un dia de la semana: ")
4
5  if (dia=="lunes"):
6      print("hoy es un dia de trabajo")
7  elif(dia=="martes"):
8      print("hoy es un dia de trabajo")
9  elif(dia=="miercoles"):
10     print("hoy es un dia de trabajo")
11  elif(dia=="jueves"):
12     print("hoy es un dia de trabajo")
13  elif(dia=="viernes"):
14     print("ya casi es fin de semana ")
15  elif(dia=="sabado"):
16     print("hoy es un dia de descanso")
17  elif(dia=="domingo"):
18     print("hoy es un dia de descanso")
19  else:
20     print("ERROR....Señor usuario no digito bien el dia de la semana")
21
```



Ejercicio 34_2.

Requerir al usuario que ingrese un día de la semana e imprimir un mensaje si es **lunes**, otro mensaje diferente si es **viernes**, otro mensaje diferente si es **sábado o domingo**. Si el día ingresado no es ninguno de esos, imprimir otro mensaje.

```
dia=input("Señor usuario digitar un día de la semana: ")
if (dia=="lunes"):
    print("Oh, no!")
elif (dia=="viernes"):
    print("¡Ya casi!")

elif (dia=="sabado" or dia=="domingo"):
    print("Ahora sí se puede descansar")
else:
    print("A esperar el fin de semana")
```

Ejercicio 35.

Escribir un programa que, dado un número entero, muestre su valor absoluto. Nota: para los números positivos su valor absoluto es igual al número (el valor absoluto de 52 es 52),



mientras que, para los negativos, su valor absoluto es el número multiplicado por -1 (el valor absoluto de -52 es 52).

```
numero=int(input("Señor usuario digitar un numero:"))
if numero<0:
    numero=numero*-1
print("Valor absoluto:", numero)
```

Función `upper()` en python.

Es una función utilizada para convertir todos los caracteres alfabéticos de la cadena en mayúscula. Todos los caracteres no alfabéticos no sufren cambios.

Ejercicio 36_1: Crear un programa que permita al usuario elegir un candidato por el cual votar.

Las posibilidades son: candidato **A por el partido rojo**, candidato **B por el partido verde**, candidato **C por el partido azul**.

Según el candidato elegido (A, B ó C) se le debe imprimir el mensaje “*Usted ha votado por el partido [color que corresponda al candidato elegido]*”. Si el usuario ingresa una opción que no corresponde a ninguno de los candidatos disponibles, indicar “*Opción errónea*”.

`.upper()`

```
candidato=input("Candidato elegido: ")
if candidato.upper()=="A":
    print("Usted ha votado por el partido rojo")
elif candidato.upper()=="B":
    print("Usted ha votado por el partido verde")
elif candidato.upper()=="C":
    print("Usted ha votado por el partido azul")
else:
    print("Opción errónea")
```

Ejercicio 36_2: Dias de la semanas aplicando la Funcion `.upper()`



```
ejercicio36_2_upper.py > ...
1  #Dias de la semana aplicando la funcion .upper()
2
3  dia=input("ingrese porfavor un dia de la semana: ")
4
5  if (dia.upper()=="LUNES"):
6      print("hoy es un dia de trabajo")
7  elif(dia.upper()=="MARTES"):
8      print("hoy es un dia de trabajo")
9  elif(dia.upper()=="MIERCOLES"):
10     print("hoy es un dia de trabajo")
11  elif(dia.upper()=="JUEVES"):
12     print("hoy es un dia de trabajo")
13  elif(dia.upper()=="VIERNES"):
14     print("ya casi es fin de semana ")
15  elif(dia.upper()=="SABADO"):
16     print("hoy es un dia d edescanso")
17  elif(dia.upper=="DOMINGO"):
18     print("hoy es un dia d edescanso")
19  else:
20     print("ERROR...señor usuario no digito bien el dia de la semana")
21
```

Ejercicio 37_1.



Algoritmo que lea una letra e identifique si es una vocal o es una consonante.

```
letra = input("Ingresa una letra: ")
if (letra == "a" or letra == "e" or letra == "i" or letra == "o" or letra == "u"):
    print ("Es vocal.")
else:
    print ("Es consonante.")
```

Ejercicio 37_1. Algoritmo que lea una letra e identifique si es una vocal o es una consonante, se requiere utilizar la función **.upper()**

```
ejercicio37_2_upper.py > ...
1  #37_2. Algoritmo que lea una letra e identifique si es una vocal o es una consonante.
2
3
4  letra = input("ingrese una letra: ")
5  if (letra.upper() == "A" or letra.upper() == "E" or letra.upper() == "I"
6      or letra.upper() == "U" or letra.upper() == "O"):
7
8      print("Es una vocal")
9
10 else:
11     print("Es una consonante")
```

Ejercicio 38.



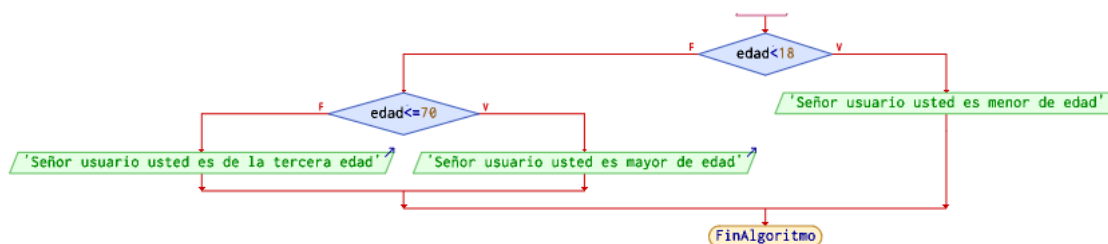
A un trabajador se le paga por sus horas trabajadas y la tarifa por hora, si la cantidad de **horas es mayor a 40 entonces el costo por la hora extra será un 50% adicional** a la tarifa por hora, calcular el salario total del trabajador.

```
ejercicio38.py > ...
7
8
9  tarifa = int(input("Ingrese la tarifa por horas: "))
10 horas = int(input("Ingrese las horas trabajadas: "))
11
12 salario = tarifa*horas
13 horasextras = horas-40
14 extras = (40*tarifa)+(horasextras*tarifa*1.5)
15
16 if horas>40:
17     print ("El salario es: ",extras)
18 else:
19     print ("El salario es:",salario)
```

Ejercicios del 39 al 47

Ejercicio 39. Determinar si una persona es mayor de edad.

Diagrama de flujo



Python

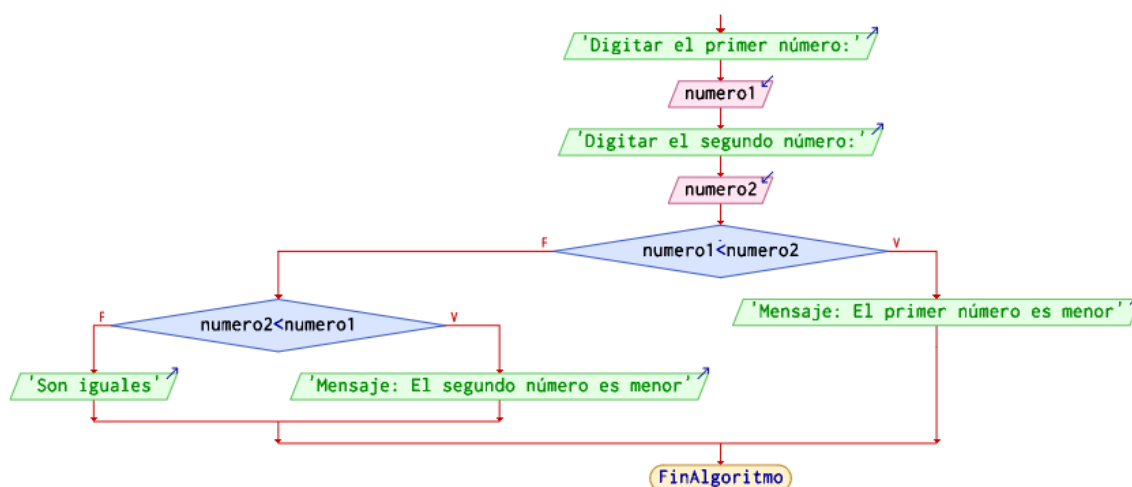
```

ejercicio39.py > ...
1  #Ejercicio 39. Determinar si una persona es mayor de edad.
2
3  edad=int(input("Ingrese su edad: "))
4
5  if (edad<18):
6      print("Señor usuario usted es menor de edad")
7  elif(edad<=70):
8      print("Señor usuario usted es mayor de edad")
9  else:
10     print("Señor usuario usted es de la tercera edad")
  
```

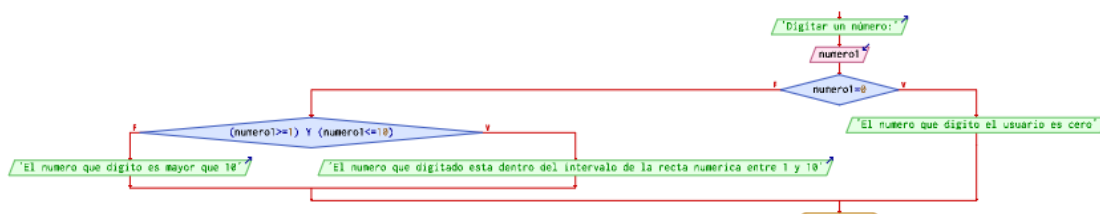
Ejercicio 40. Comparación. Solicitar un número al usuario y mostrar el resultado **multiplicado por cien (100)**, sólo en caso de que sea mayor de 500.



Ejercicio 41. Comparación. Solicitar al usuario que ingrese dos números, ilustrando cuál de los dos números es menor o son iguales.

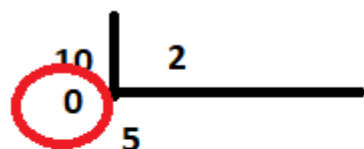


Ejercicio 42. Estructura de Selección múltiples. Solicitar un número al usuario y **mostrar si es cero, se encuentra entre 1 y 10** o éste **es mayor de 10**.

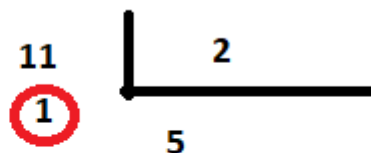


Ejercicio 43. Se desea saber si un número es par.

Tener presente:

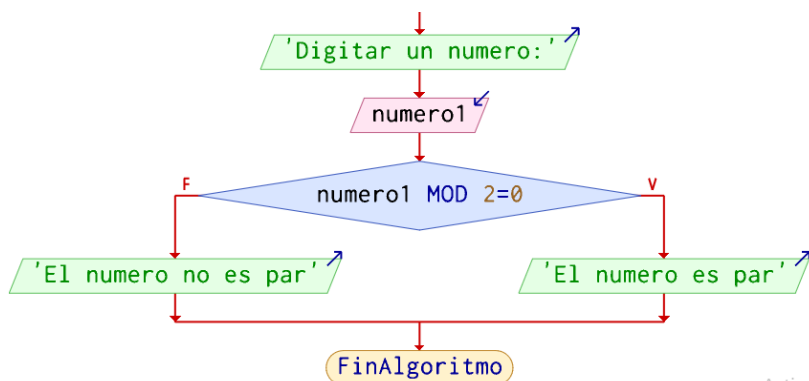


**Si el residuo = 0
el numero es PAR**



**Si el residuo =1
el numero es IMPAR**

Diagrama de Flujo



Activar

Python: Modulo %

```
ejercicio43_1.py > ...
1  #Se desea saber si un número es par.
2
3  numero=int(input("Señor usuario digitar un numero:"))
4
5  resultado=numero%2
6
7  print("Residuo de la division es igual a:", resultado)
8
9  if resultado==0:
10     print(f"El numero es par: {resultado}")
11
12 else:
13     print(f"El numero es impar: {resultado}")
14
```

Otra ejemplo:

```
ejercicio43_2.py > ...
1  #Se desea saber si un número es par.
2
3  numero=int(input("Señor usuario digitar un numero:"))
4
5  if (numero%2 == 0):
6     print("El numero digitado es par", (numero))
7  else :
8     print("El numero digitado es impar", (numero))
```

Ejercicio 44. Valor de la llamada. Determinar la cantidad total a pagar por una llamada telefónica, teniendo en cuenta lo siguiente:

- Toda llamada que **dure menos de tres minutos** tiene un costo de \$10000.
- Cada minuto adicional a partir de los tres minutos **cuesta \$1500**.

Ejercicio 45. Si gasto hasta \$100.000, pago con dinero en efectivo. Si no, si gasto más de \$100.000 pero menos de \$300.000, pago con tarjeta de débito. Si no, pago con tarjeta de crédito.



Ejercicio 46. Se le informa a un cliente, que si la compra es mayor a \$100.000, obtiene un descuento del 10%, de lo contrario informar con un mensaje que no tiene ningún descuento.

Ejercicio 47. Escribe un programa que solicite una puntuación entre 0.0 y 1.0. Si la puntuación está fuera de ese rango, muestra un mensaje de error. Si la puntuación está entre 0.0 y 1.0, muestra la calificación usando la tabla siguiente:

Puntuación	Calificación
≥ 0.9	Sobresaliente
≥ 0.8	Notable
≥ 0.7	Bien
≥ 0.6	Suficiente
< 0.6	Insuficiente

Prueba de escritorio: Ejecuta el programa repetidamente, para probar los siguientes valores de entrada.

Introduzca puntuación: 0.95
Sobresaliente

Introduzca puntuación: perfecto
Puntuación incorrecta

Introduzca puntuación: 10.0
Puntuación incorrecta

Introduzca puntuación: 0.75
Bien

Introduzca puntuación: 0.5
Insuficiente



Material de Referencia

- Cómo usar ciclos en Python, <https://codigonaranja.com/ciclos-en-python>



Este material puede ser distribuido, copiado y exhibido por **terceros si se muestra en los créditos**. No se puede obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de la licencia que el documento (instructivo) original.