



**UNIVERSIDAD
DEL VALLE
DE ORIZABA**

UnivO®

Universidad del Valle de Orizaba

EXAMEN

Miguel Angel Tress Ruiz

Docente. Erik Nuñez Delgado

**Licenciatura en Ingeniería en Sistemas
computacionales**

Asignatura. Matemáticas Discretas





EJERCICIO 1

CALCULO DE INDICE DE MASA CORPORAL

Como Ingeniero en Sistemas Computacionales con bases en Matemáticas Discretas, deberás recordar que el portal de:

en su implementación existen algunos puntos a mejorar:

- La captura de datos de estatura no valida el teclear números negativos o el cero.
- La captura de datos de peso no valida el teclear números negativos o el cero.
- El dominio de los valores para estatura apegados a la realidad no está contemplado.
- El dominio de los valores para peso apegados a la realidad no está contemplado.
- El dato del valor 18.5 no está del todo claro.

1. Primero que nada, lo que haremos es declarar las variables comenzando nuestro código de las cuales serán las siguientes:

```
# Inicialización de variables
peso = 0.0
estatura = 0.0
imc = 0.0
```

2. Crearemos una condición con while donde este nos dirá que se debe ingresar una estatura que se encuentre en el rango que se menciona este bucle se va cerrar solo cuando se ingrese un dato valido en caso de no serlo se arrojará un mensaje para que se realice de manera correcta, en caso de ingresar un valor no numérico se realizara ValueError y arrojará otro mensaje

```
while True:
    try:
        estatura = float(input("Teclea estatura en metros (entre 1.40 y 2.20): "))
        if 1.40 <= estatura <= 2.20:
            break
        else:
            print("La estatura debe estar entre 1.40 y 2.20 metros.")
    except ValueError:
        print("Por favor, teclea un número válido.")
```

3. Por consiguiente, haremos lo mismo con el peso sin embargo por obvias razones se debe cambiar el rango y que ahora estamos utilizando KG



```
while True:
    try:
        estatura = float(input("Teclea estatura en metros (entre 1.40 y 2.20): "))
        if 1.40 <= estatura <= 2.20:
            break
        else:
            print("La estatura debe estar entre 1.40 y 2.20 metros.")
    except ValueError:
        print("Por favor, teclea un número válido.")
```

4. Después de agregar las condiciones que validaran los datos que se van a ingresar colocaremos la función que será la encargada de calcular el IMC con los datos obtenidos mediante el usuario

3

```
# Cálculo del IMC
imc = peso / (estatura ** 2)
print("\n El IMC es:", round(imc, 2))
```

5. Por último, se agregaría la clasificación del IMC donde evaluara el resultado obtenido con las diferentes composiciones corporales que hay que son: Peso inferior al normal, normal, pero superior al normal y obesidad

```
# Clasificación del IMC
if imc < 18.5:
    print("Composición corporal: Peso inferior al normal")
elif 18.6 <= imc <= 24.9:
    print("Composición corporal: Normal")
elif 25.0 <= imc <= 29.9:
    print("Composición corporal: Peso superior al normal")
else:
    print("Composición corporal: Obesidad")
```

Programa en funcionamiento

```
Teclea estatura en metros (entre 1.40 y 2.20): 1.30
La estatura debe estar entre 1.40 y 2.20 metros.
Teclea estatura en metros (entre 1.40 y 2.20): -20
La estatura debe estar entre 1.40 y 2.20 metros.
Teclea estatura en metros (entre 1.40 y 2.20): g
Por favor, teclea un número válido.
Teclea estatura en metros (entre 1.40 y 2.20):
```

Estatura 1



```
Teclea estatura en metros (entre 1.40 y 2.20): 1.50
Teclea peso en kilogramos (entre 20 y 120): 10
El peso debe estar entre 20 kg y 120 kg.
Teclea peso en kilogramos (entre 20 y 120): -8
El peso debe estar entre 20 kg y 120 kg.
Teclea peso en kilogramos (entre 20 y 120): ñ
Por favor, teclea un número válido.
Teclea peso en kilogramos (entre 20 y 120):
```

Peso 1

```
Teclea peso en kilogramos (entre 20 y 120): 65

El IMC es: 28.89
Composición corporal: Peso superior al normal
```

Resultado 1

EJERCICIO 2

CONJUNTOS VÍA DICIONARIOS DE PYTHON

Del portal de Sistemas de Consulta del INEGI, se obtuvieron los accidentes por estado, así como el número de Heridos y Muertes

Como Analista de Datos, se pide que respondas lo siguiente:

¿Cuál es el estado y la cantidad con mayor número de accidentes? Este punto está resuelto para que sea tomado como referencia.

¿Cuál es el estado y la cantidad con menor número de accidentes?

¿Cuál es el estado y la cantidad con mayor número de heridos?

¿Cuál es el estado y la cantidad con menor número de heridos?

¿Cuál es el estado y la cantidad con mayor número de muertes?

¿Cuál es el estado y la cantidad con menor número de muertes?

¿Cuál es el total de Accidentes?

¿Cuál es el total de heridos?

¿Cuál es el total de muertes?

Completar el ciclo donde se muestra el listado de estados y las cantidades de Accidentes con el porcentaje por cada estado. En la función print debes agregar la operación del cálculo del porcentaje.

```
1 import operator
2
```

Se debe de importar el modulo operator para poder usar operator.itemgetter(1), el cual este permite comparar los valores de los diccionarios en lugar de las llaves. Este es algo útil para encontrar al máximo o mínimo según el valor numérico.

5

```
# Diccionarios de datos
Accidentes = {
    'AGU':4473,'BCN':11782,'BCS':6045,'CAM':4574,'COA':10876,'COL':6225,'CHP':4065,'CHH':26559,'DIF':7276,
    'DUR':11631,'GUA':17703,'GRO':7414,'HID':4593,'JAL':14293,'MEX':21105,'MIC':16196,'MOR':8864,'NAY':2852,
    'NLE':80582,'OAX':5203,'PUE':13423,'QUE':9307,'ROO':6619,'SLP':6775,'SIN':7230,'SON':26557,'TAB':2492,
    'TAM':14090,'TLA':1674,'VER':9659,'YUC':8478,'ZAC':2433
}

Heridos = {
    'AGU':1269,'BCN':2210,'BCS':1292,'CAM':1086,'COA':3776,'COL':1843,'CHP':1153,'CHH':6540,'DIF':2431,
    'DUR':4069,'GUA':6549,'GRO':2298,'HID':796,'JAL':4021,'MEX':4016,'MIC':4439,'MOR':2657,'NAY':1063,
    'NLE':5611,'OAX':1715,'PUE':2461,'QUE':1557,'ROO':3593,'SLP':1133,'SIN':4972,'SON':3825,'TAB':931,
    'TAM':2926,'TLA':386,'VER':3403,'YUC':5469,'ZAC':1010
}

Muertes = {}
    'AGU':93,'BCN':70,'BCS':36,'CAM':54,'COA':108,'COL':43,'CHP':108,'CHH':292,'DIF':298,'DUR':113,
    'GUA':183,'GRO':111,'HID':50,'JAL':335,'MEX':482,'MIC':283,'MOR':79,'NAY':77,'NLE':255,'OAX':85,
    'PUE':282,'QUE':181,'ROO':88,'SLP':109,'SIN':289,'SON':232,'TAB':28,'TAM':161,'TLA':79,'VER':68,
    'YUC':57,'ZAC':74
}
```

Como siguiente paso definimos tres diccionarios (accidentes, heridos, muertes) en donde en cada uno contiene la información de los estados con mayor numero de los aspectos que se mencionan, todo esto durante el año 2023.

```
# Funciones para obtener mayor y menor por diccionario
def obtener_extremos(dic, descripcion):
    max_estado = max(dic.items(), key=operator.itemgetter(1))
    min_estado = min(dic.items(), key=operator.itemgetter(1))
    print(f"Estado con MAYOR número de {descripcion}: {max_estado[0]} - {max_estado[1]}")
    print(f"Estado con MENOR número de {descripcion}: {min_estado[0]} - {min_estado[1]}")
    return sum(dic.values())
```

Después declaramos una función llamada obtener_extremos en la cual esta va a tener dos parámetros que son dic que es nuestro diccionario y descripción que es para que tenga sentido. En la siguiente línea declaramos max_estado donde este se encargará de encontrar el valor más alto, ingresamos el método dic.items() este se encargará de convertir el diccionario en una lista de pares (clave, valor) y operator.itemgetter(1) le dice a max que compare el segundo valor par

que es el valor numérico no el clave; esta buscara el par con el valor más alto. En el caso del min es lo mismo solo que aquí buscara el valor más bajo.

Por último en las últimas líneas se encargarán de imprimir el estado con mayor numero de: accidente, muertes o heridos (max_estado[0] – max_estado[1], min_estado[0] – min_estado[1] es la clave y el valor).

```
return sum(dic.values())
```

Este se encarga de sumar todos los valores del diccionario y devolverlo

```
# Accidentes
print("\n--- Accidentes ---")
total_accidentes = obtener_extremos(Accidentes, "accidentes")

# Heridos
print("\n--- Heridos ---")
total_heridos = obtener_extremos(Heridos, "heridos")

# Muertes
print("\n--- Muertes ---")
total_muertes = obtener_extremos(Muertes, "muertes")
```

En este apartado de nuestro código se encarga de imprimir el mayor estado y menor estado con numero de: accidentes, heridos y muertes, esto mediante una llamada a la función que se hizo anteriormente y la suma total lo guarda en total_accidentes, total_heridos y total_muertes. Este proceso se guarda en los tres casos

```
# Totales nacionales
print("\n--- Totales Nacionales ---")
print(f"Total de Accidentes: {total_accidentes}")
print(f"Total de Heridos: {total_heridos}")
print(f"Total de Muertes: {total_muertes}")
```

Como últimos bloques de código se encuentra este que se encarga de imprimir solamente la suma para categoría, esto mandando a traer la función que hizo anteriormente

```
# Porcentajes por estado
print("\n--- Porcentaje de Accidentes por Estado ---")
for estado, cantidad in Accidentes.items():
    porcentaje = (cantidad / total_accidentes) * 100
    print(f"Estado: {estado} - Accidentes: {cantidad} - Porcentaje: {porcentaje:.2f}%")
```

Este bloque se encarga de calcular el porcentaje de accidentes de cada estado con respecto a la suma total de accidentes

- Recorre cada estado en el diccionario Accidentes.
- Calcula el porcentaje que representa ese estado con respecto al total nacional de accidentes.
- Muestra ese porcentaje con dos decimales usando `:.2f`.

Programa en ejecución

```
--- Accidentes ---
Estado con MAYOR número de accidentes: NLE - 80582
Estado con MENOR número de accidentes: TLA - 1674

--- Heridos ---
Estado con MAYOR número de heridos: GUA - 6549
Estado con MENOR número de heridos: TLA - 386

--- Muertes ---
Estado con MAYOR número de muertes: MEX - 482
Estado con MENOR número de muertes: TAB - 28

--- Totales Nacionales ---
Total de Accidentes: 381048
Total de Heridos: 90500
Total de Muertes: 4803
```

```
--- Porcentaje de Accidentes por Estado ---
Estado: AGU - Accidentes: 4473 - Porcentaje: 1.17%
Estado: BCN - Accidentes: 11782 - Porcentaje: 3.09%
Estado: BCS - Accidentes: 6045 - Porcentaje: 1.59%
Estado: CAM - Accidentes: 4574 - Porcentaje: 1.20%
Estado: COA - Accidentes: 10876 - Porcentaje: 2.85%
Estado: COL - Accidentes: 6225 - Porcentaje: 1.63%
Estado: CHP - Accidentes: 4065 - Porcentaje: 1.07%
Estado: CHH - Accidentes: 26559 - Porcentaje: 6.97%
Estado: DIF - Accidentes: 7276 - Porcentaje: 1.91%
Estado: DUR - Accidentes: 11631 - Porcentaje: 3.05%
Estado: GUA - Accidentes: 17703 - Porcentaje: 4.65%
Estado: GRO - Accidentes: 7414 - Porcentaje: 1.95%
Estado: HID - Accidentes: 4593 - Porcentaje: 1.21%
Estado: JAL - Accidentes: 14293 - Porcentaje: 3.75%
Estado: MEX - Accidentes: 21105 - Porcentaje: 5.54%
Estado: MIC - Accidentes: 16196 - Porcentaje: 4.25%
Estado: MOR - Accidentes: 8864 - Porcentaje: 2.33%
Estado: NAY - Accidentes: 2852 - Porcentaje: 0.75%
Estado: NLE - Accidentes: 80582 - Porcentaje: 21.15%
Estado: OAX - Accidentes: 5203 - Porcentaje: 1.37%
Estado: PUE - Accidentes: 13423 - Porcentaje: 3.52%
Estado: QUE - Accidentes: 9307 - Porcentaje: 2.44%
Estado: ROO - Accidentes: 6619 - Porcentaje: 1.74%
Estado: SLP - Accidentes: 6775 - Porcentaje: 1.78%
Estado: SIN - Accidentes: 7230 - Porcentaje: 1.90%
Estado: ROO - Accidentes: 6619 - Porcentaje: 1.74%
Estado: SLP - Accidentes: 6775 - Porcentaje: 1.78%
Estado: SIN - Accidentes: 7230 - Porcentaje: 1.90%
Estado: SIN - Accidentes: 7230 - Porcentaje: 1.90%
Estado: SON - Accidentes: 26557 - Porcentaje: 6.97%
Estado: TAB - Accidentes: 2492 - Porcentaje: 0.65%
Estado: TAM - Accidentes: 14090 - Porcentaje: 3.70%
Estado: TLA - Accidentes: 1674 - Porcentaje: 0.44%
Estado: VER - Accidentes: 9659 - Porcentaje: 2.53%
Estado: TAB - Accidentes: 2492 - Porcentaje: 0.65%
Estado: TAM - Accidentes: 14090 - Porcentaje: 3.70%
Estado: TLA - Accidentes: 1674 - Porcentaje: 0.44%
Estado: VER - Accidentes: 9659 - Porcentaje: 2.53%
Estado: VER - Accidentes: 9659 - Porcentaje: 2.53%
Estado: YUC - Accidentes: 8478 - Porcentaje: 2.22%
```