

# ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERIA EN ELECTRICIDAD Y COMPUTACION

## PRACTICA AUTÓNOMA UNIDAD 3 – ESTRUCTURAS DE CONTROL E ITERATIVAS

### FUNDAMENTOS DE PROGRAMACIÓN

#### IMPORTANTE: LEER DETENIDAMENTE ANTES DE COMENZAR EL TALLER

El programa calculará y mostrará información para una empresa que alquila vehículos a sus clientes. Para un cliente especificado, el programa calculará y mostrará la cantidad de dinero cobrado por el alquiler de vehículos de ese cliente.

#### Especificaciones:

El programa está dividido en opciones que son:

1. Ingreso de datos: el programa debe solicitar continuamente al usuario que introduzca la siguiente información para cada cliente (en el orden especificado):
  - a. El código de clasificación del cliente (un carácter, siempre será en mayúscula)
  - b. El número de días de alquiler del vehículo (un número entero)
  - c. La lectura inicial del odómetro del vehículo, es decir al comienzo del período de alquiler (un número entero)
  - d. La lectura final del odómetro del vehículo, es decir al final del período de alquiler (un número entero)Luego de cada ingreso del cliente y alquiler del vehículo el sistema preguntará si desea continuar ingresando información. Si no desea continuar finaliza esta opción.
2. El programa debe calcular el número de kilómetros conducidos por cada cliente durante el periodo de alquiler.
3. El programa debe calcular el costo del alquiler que se factura a un cliente. Basado en el *código de clasificación del cliente, número de días en el período de alquiler y número de kilómetros recorridos*. Para este cálculo debe utilizar a, b, y c que se describen:
  - a. Código "E" (Económico)  
Cargo base: \$ 40.00  
Cargo por kilometraje: \$ 0.25 por cada kilómetro recorrido
  - b. Código "D" (diario)  
Cargo base: \$ 60.00  
Cargo de kilometraje: sin cargo si el promedio de kilómetros recorridos por día es de 100 kilómetros o menos; De lo contrario, \$ 0.25 por cada kilómetro conducido por encima del límite de 100 kilómetros por día.
  - c. Código "S" (semanal)  
Cargo base: \$ 190.00  
Cargo de kilometraje: \$0 si el promedio de kilómetros recorridos por semana es de 900 kilómetros o menos; \$ 100.00 por semana si el promedio de kilómetros recorridos por semana excede los 900 kilómetros, pero no exceda de 1500 kilómetros; De lo contrario,

\$ 200.00 por semana más \$ 0.25 por cada kilómetro conducido sobre el límite de 1500 kilómetros por semana.

El importe facturado al cliente es la suma del cargo base más el cargo por kilometraje.

4. Para cada cliente, el programa mostrará un resumen con la siguiente información:
  - a. El código de clasificación del cliente
  - b. El número de días que el vehículo fue rentado
  - c. El odómetro del vehículo al comienzo del periodo de renta
  - d. El odómetro del vehículo al final del periodo de renta
  - e. El número de kilómetros manejados durante el periodo de renta.
  - f. La cantidad de dinero facturada al cliente en el periodo de renta.
5. Al salir del programa el programa debe mostrar la suma en dólares de todos los alquileres calculados en el sistema.
6. Validar los códigos de clasificación para que solo se acepten las opciones definidas en el punto 3.

### Elaboración:

Después de tener una idea clara del objetivo del programa, lo más importante es descomponer e identificar piezas claves del programa que deseamos elaborar. No es recomendable resolver todo en un solo paso.

Comenzaremos con el punto 1:

En este punto se identifica que el usuario requerirá el ingreso de 4 atributos, con los cuales el programa completará el objetivo (ver Figura 1).

```
4  #<<Ingreso de Datos>>
5  codClasif = input("Ingrese codigo de clasificacion: ").upper()
6  numeroDias = int(input("Numero de dias de alquiler: "))
7  odometroInicio = int(input("Odometro Inicial: "))
8  odometroFin = int(input("Odometro Final: "))
```

Figura 1: Ingreso de Datos

El atributo código de clasificación (a), es un carácter que identifica el tipo de alquiler, y será siempre en mayúsculas, por ello luego de la sentencia input lo transformamos a mayúscula con la función .upper(), de una forma más directa. Otra opción es la siguiente:

```
codClasif = input("Ingrese codigo de clasificacion: ")
CodClasif = codClasif.upper()
```

Para los atributos (b), (c) y (d), el usuario debe ingresar números enteros, por ello convertimos los datos ingresados con la sentencia input, con la función int(). Recordemos que todos los datos ingresados con el input por default son de tipo String y deben ser transformados para que sean reconocidos como números.

Hasta aquí, lo único que hemos logrado es que el usuario ingrese los cuatros datos requeridos. Sin embargo, el programa requiere que el usuario pueda ingresar datos mientras desee continuar en el programa. Para lograr este objetivo, añadiremos un input adicional (ver Figura 2):

```

4      #<<Ingreso de Datos>>
5      codClasif = input("Ingresa codigo de clasificacion: ").upper()
6      numeroDias = int(input("Numero de dias de alquiler: "))
7      odometroInicio = int(input("Odometro Inicial: "))
8      odometroFin = int(input("Odometro Final: "))
9
10     continuar = input("Desea continuar? S/N").upper()

```

Figura 2: Creación Variable continuar

En la línea 10 (Figura 2) solicitamos al usuario responder la pregunta, si desea o no continuar en el programa. A través del `input()` y la variable **continuar**. Por si sola esta variable representa solamente un ingreso por teclado. Si el usuario ingresa "S" significa que si desea continuar con el programa y "N" lo contrario.

En este punto, podemos detectar que hay un ciclo que se ejecutará mientras una condición sea verdadera. Es decir, si la variable **continuar** es "S" debe continuar en el ciclo, caso contrario si es "N" (diferente de "S") no continua en el ciclo. Si el usuario no desea seguir, el programa ya no deberá pedirle ingresar más datos. Para lograr esto usaremos la sentencia **while** (ver Figura 3):

```

2      continuar=""
3      while continuar != "N":
4          #<<Ingreso de Datos>>
5          codClasif = input("Ingresa codigo de clasificacion: ").upper()
6          numeroDias = int(input("Numero de dias de alquiler: "))
7          odometroInicio = int(input("Odometro Inicial: "))
8          odometroFin = int(input("Odometro Final: "))
9
10         continuar = input("Desea continuar? S/N").upper()

```

Figura 3: Creación de un ciclo

Examinemos el código anterior, en la línea 3 (Figura 3) vemos que la condición del **while** es que **continuar** sea diferente de la letra "N". Es decir, que si el usuario ingresa "S", el programa seguirá ejecutándose. También es importante darse cuenta que la indentación de nuestro código cambio. Todos los `inputs` se han movido 4 espacios a la derecha, lo que nos indica que ahora el ingreso de dato está contenido en un ciclo.

Adicionalmente notemos que, en la línea 2 del código previo estamos iniciando la variable **continuar** con un valor vacío (""). La razón es que la variable **continuar** debe estar definida previamente para ser usada en el **while**.

Trata de comprobarlo en tu IDE, al remover la definición de la variable **continuar** (línea 2), y ejecutar el programa se presentará un error como el siguiente (ver Figura 4):

```
Traceback (most recent call last):
  File "C:/Users/Luci/PycharmProjects/untitled1/alquiler.py",
    while continuar != "N":
NameError: name 'continuar' is not defined
```

Figura 4: Error no definición de variable

Es de mucha importancia irse familiarizando con este tipo de errores y saber cómo solucionarlos. Prueba la ejecución y asegúrate que el ingreso de datos esté funcionando correctamente antes de continuar.

En el punto 2 debemos calcular el kilometraje para ello nos valemos de las variables **odometroInicio** y **odometroFin**. La diferencia de estas variables es el cálculo de kilometraje (línea 11) (ver Figura 5).

Para el paso 3 necesitamos calcular el costo de alquiler a pagar por cliente y este depende del código de clasificación que se haya ingresado. Esto significa que está condicionado a lo que ingresa el usuario y del valor que está almacenado en la variable **codClasif** del paso 1. Para ello utilizaremos la sentencia **if** como se observa en la línea 14 (Figura 5), donde comenzamos la evaluación de la variable para saber si es equivalente a "E" lo cual significa que el usuario escogió un alquiler "Económico"

```
3 while continuar != "N":
4     #<<Ingreso de Datos>>
5     codClasif = input("Ingrese codigo de clasificacion: ").upper()
6     numeroDias = int(input("Numero de dias de alquiler: "))
7     odometroInicio = int(input("Odometro Inicial: "))
8     odometroFin = int(input("Odometro Final: "))
9
10    #<<Logica de Calculo>>
11    kilometraje = odometroFin - odometroInicio
12    cargoBase = 0
13    cargoKm = 0
14    if codClasif=="E":...
18    elif codClasif=="D":...
24    else:...
```

Figura 5: Calculo de Kilometraje

Es importante notar, que tenemos una instrucción del tipo **if-elif-else** y no tenemos instrucciones **if** independientes. La razón es que el cálculo del costo de alquiler depende de cada código de clasificación (E,D,S) y solo es posible una forma de cálculo a la vez. Por el contrario, al tener **if** independientes, se puede producir el problema de aplicar varios cálculos, si más de una condición se vuelve verdadera.

Optimo	No Optimo
<b>if</b> codClasif=="E": ...	<b>if</b> codClasif=="E":
<b>elif</b> codClasif=="D": ...	...
<b>else:</b>	<b>if</b> codClasif=="D":
	...
	<b>if</b> codClasif == "S":
	...

Para comenzar la evaluación, empezamos por el código de clasificación “E” (Económico), en este caso simplemente tenemos un cargo base de 40 y el cargo por kilometraje igual al número de kilómetros calculados previamente multiplicado por un factor de 0.25 (línea 16)(Figura 6).

```
11      #<<Logica de Calculo>>
12      kilometraje = odometroFin - odometroInicio
13      if codClasif=="E":
14          cargoBase=40
15          factorKM=0.25
16          cargoKm=kilometraje*factorKM
17      elif codClasif=="D":...
23      else:...
```

Figura 6: Calculo de Alquiler económico

En el código de clasificación “D”, el cargo por kilometraje depende del promedio de kilómetros por día, si es menor a 100 no habrá cargo por kilómetros y solo se cobrará el cargo base. Podemos identificar en este caso que existe una instrucción condicional dentro de otra lo que se conoce como el nombre de **if anidado**. En la línea 20 podemos ver la segunda instrucción if con la condición del promedio mayor a 100. Observe que el bloque de instrucciones para calcular el cargo por kilometraje (líneas 21 y 22)(ver Figura 7) está indentado a la derecha del condicional, lo cual indica que pertenece a la instrucción **if** de la línea 20.

```
11      #<<Logica de Calculo>>
12      kilometraje = odometroFin - odometroInicio
13      if codClasif=="E":...
17      elif codClasif=="D":
18          cargoBase = 60
19          #Calculamos el promedio
20          if kilometraje/numeroDias > 100:
21              factorKM = 0.25
22              cargoKm = kilometraje * factorKM
23      else:...
```

Figura 7: Calculo de Alquiler Diario

Por último, el código de clasificación semanal “S” (ver Figura 8) evalúa el promedio semanal en varios rangos. Luego de establecer el cargo base (línea 24) obtenemos el cargo por kilometraje. Para el cálculo por kilometraje, primero calculamos el número de semanas dividiendo el número de días ingresados para 7 (línea 25). Luego, obtenemos el kilometraje semanal lo almacenamos en la variable **kmSemana** (línea 26). Si cumple un rango mayor a 900 kilómetros hasta los 1500 kilómetros se calculará de una forma. De lo contrario, el cargo será diferente. Puedes revisar en la especificación las diferencias que existen en los calculos.

```

11      #<<Logica de Calculo>>
12      kilometraje = odometroFin - odometroInicio
13      if codClasif=="E":...
17      elif codClasif=="D":...
23      else:
24          cargoBase = 190
25          semanas = int(numeroDias/7)
26          kmSemana = kilometraje/ semanas
27          if kmSemana >900 and kmSemana<=1500:
28              cargoKm = 100* semanas
29          elif kmSemana>1500:
30              cargoKm=(200*semanas)+( (kmSemana-1500) *0.25)

```

Figura 8 Calculo Semanal

Notemos también que en este caso en la **línea 23** (ver Figura 8) estamos usando directamente un **else**, dado que el código de clasificación solo tiene 3 posibles valores el último valor posible en teoría es la **"S"**, por ende simplemente usamos **else**, porque hemos evaluado previamente el código de clasificación **"E"** y **"D"**

Para el paso 4 (ver Figura 9), tenemos que presentar la información referente al alquiler al usuario. Recordemos que el total a cancelar es la suma del cargo base más el cargo por kilómetros. Nótese en el código siguiente, que el cálculo del total esta fuera del **if** (pero dentro del **while**) puesto que una vez obtenido el cargo base y por kilometraje la suma y la presentación de resultados es independiente de la evaluación del código de clasificación.

```

11      #<<Logica de Calculo>>
12      kilometraje = odometroFin - odometroInicio
13      if codClasif=="E":...
17      elif codClasif=="D":...
23      else:...
31
32      total = cargoBase+cargoKm
33      print("Codigo de Clasificacion:", codClasif)
34      print("Numero de dias de alquiler:", numeroDias)
35      print("Odometro Inicial:", odometroInicio, "Odometro Final:", odometroFin)
36      print("Numero de Kilometros:", kilometraje)
37      print("Total a Pagar:",total )

```

Figura 9 Presentación de datos

El paso 5, nos indica que al salir del programa debe presentarse al usuario el total de todos los alquileres calculados. Para ello crearemos un acumulador que es una variable a la cual se irá sumando cada uno de los valores totales. En la línea 41 vemos que la variable **factAcum** se irá incrementando con los resultados de la variable total. Esta operación se encuentra dentro del ciclo **while** (línea 3), es decir por cada iteración en el lazo la variable acumuladora irá incrementándose.

En la línea 44 (ver Figura 10) se usa la función **print()** para imprimir el total acumulado en la variable **factAcum**. Nótese que esta línea no está indentada, es decir que esta fuera del **while**. Por ende, se va a ejecutar cuando el usuario haya decidido salir del ciclo.



```

1  continuar=""
2  factAcum = 0
3  while continuar != "N":
4      cargoBase = 0
5      cargoKm = 0
6      #<<Ingreso de Datos>>
7      codClasif = input("Ingrese codigo de clasificacion: ").upper()
8      numeroDias = int(input("Numero de dias de alquiler: "))
9      odometroInicio = int(input("Odometro Inicial: "))
10     odometroFin = int(input("Odometro Final: "))
11
12     #<<Logica de Calculo>>
13     kilometraje = odometroFin - odometroInicio
14     if codClasif=="E":...
18     elif codClasif=="D":...
24     else:...
32
33     #<<Impresion de Resultados>>
34     total = cargoBase+cargoKm
35     print("Codigo de Clasificacion:", codClasif)
36     print("Numero de dias de alquiler:", numeroDias)
37     print("Odometro Inicial:", odometroInicio, "Odometro Final:", odometroFin)
38     print("Numero de Kilometros:", kilometraje)
39     print("Total a Pagar:",total )
40
41     factAcum = factAcum+ total
42     continuar= input("Desea continuar? S/N").upper()
43
44     print("Total Alquileres General",factAcum)

```

Figura 10 Acumulador Total

Hasta aquí el programa debería funcionar correctamente, antes de continuar con el último punto asegúrate que no tengas errores y que todo marcha bien.

El paso 6, busca asegurar el correcto funcionamiento de tu programa a través de validaciones. Recuerdas en la evaluación del código de clasificación semanal “S”, mencionamos que en teoría el usuario debe ingresar cualquiera de las 3 opciones validas: “E”, “D” o “S”, pero en la realidad nada le impide ingresar otra letra, cadena, símbolo o número. Por ello vamos a validar en este caso que nuestro usuario solo pueda ingresar valores correctos, mientras no ingrese un carácter valido, el sistema continuamente le pedirá que ingrese otro valor. ¡Fácilmente podemos detectar un ciclo! Cambiaremos un poco el input inicial y agregaremos el siguiente **while** (ver Figura 11):

```

5  codClasif = ""
6  while codClasif != "E" and codClasif != "S" and codClasif != "D":
7      codClasif = input("Ingrese codigo de clasificacion: ").upper()

```

Figura 11 Validación Códigos de clasificación

El **while** valida que mientras la variable **codClasif** sea diferente de “E”, “S” y “D” el sistema le pedirá al usuario que ingrese otro valor.

Con esto hemos completado el taller correspondiente a la unidad 3, el código completo lo puedes ver a continuación (ver Figura 12).

```
1 factAcum = 0
2 continuar=""
3 while continuar != "N":
4     #<<Ingreso de Datos>>
5     codClasif = ""
6     while codClasif != "E" and codClasif != "S" and codClasif != "D":
7         codClasif = input("Ingrese codigo de clasificacion: ").upper()
8     numeroDias = int(input("Numero de dias de alquiler: "))
9     odometroInicio = int(input("Odometro Inicial: "))
10    odometroFin = int(input("Odometro Final: "))
11
12    #<<Logica de Calculo>>
13    kilometraje = odometroFin - odometroInicio
14    cargoBase = 0
15    cargoKm = 0
16    if codClasif=="E":
17        cargoBase=40
18        factorKM=0.25
19        cargoKm=kilometraje*factorKM
20    elif codClasif=="D":
21        cargoBase = 60
22        #Calculamos el promedio
23        if kilometraje/numeroDias > 100:
24            factorKM = 0.25
25            cargoKm = kilometraje * factorKM
26    else:
27        cargoBase = 190
28        semanas = int(numeroDias/7)
29        kmSemana = kilometraje/ semanas
30        if kmSemana >900 and kmSemana<=1500:
31            cargoKm = 100* semanas
32        elif kmSemana>1500:
33            cargoKm=(200*semanas)+((kmSemana-1500)*0.25)
34
35    #<<Impresion de Resultados>>
36    total = cargoBase+cargoKm
37    print("Codigo de Clasificacion:", codClasif)
38    print("Numero de dias de alquiler:", numeroDias)
39    print("Odometro Inicial:", odometroInicio, "Odometro Final:",
40          odometroFin)
41    print("Numero de Kilometros:", kilometraje)
42    print("Total a Pagar:",total )
43
44    factAcum = factAcum+ total
45    continuar= input("Desea continuar? S/N").upper()
46 print("Total Alquileres General",factAcum)
```

Figura 12 Código Completo