



Ciclo 1 Fundamentos de programación

Reto 3 – Sistematizar las veces que un cliente entra a los supermercados “metro”.

Descripción del problema: Metro es una cadena de supermercados de origen peruano, perteneciente al holding chileno Cencosud. Tiene presencia en Perú desde 1992 y Colombia desde 2013. Actualmente, la marca Metro cuenta con más de 70 locales a nivel nacional (hipermercados y supermercados), convirtiéndola en la marca más grande de Cencosud en el Perú. Entre 2011 y 2013, el logo fue modificado: cambió al color rojo y aplicó los nuevos estándares de Cencosud. Desde 2017 hasta 2018, la cantante peruana Maricarmen Marín apareció en la publicidad de Metro, como parte de la campaña "En Metro yo confío". Metro se caracteriza por tener el conteo de cuantas veces llega un cliente en una semana, por esto ha decidido contratar a un ingeniero de Sistemas e informática para realizar un algoritmo en Python que agrupe las veces que un cliente ha ingresado y mostrar la información de dos formas distintas: una en la que mostrará el total de veces ingresadas en toda la semana y la última mostrará el número de veces seguida que vino en la semana.

Para realizar el ejercicio, se realizarán tres funciones, las cuales son:

Primera Función: Esta función recibe como parámetro único una lista con los clientes. Esta función debe retornar en una lista la cantidad de veces que se nombra en la lista, esta información se guardará en una tupla en la lista de la siguiente manera: (cliente, Nveces).

Segunda Función: Esta función recibe como parámetro único una lista con los clientes. Esta función debe retornar en una lista la cantidad de veces seguidas que el cliente ha llegado, esta información se guardará en una tupla en la lista de la siguiente manera: (cliente, Nveces).

Tercera Función: Esta función recibe como parámetros, una lista generada ya sea por la Primera Función o la Segunda Función, y un número que dará la opción a imprimir, este número es para saber si deseamos imprimir la Primera Función o la Segunda Función. Esta función debe retornar la información de las tuplas de dos maneras: Primer caso (1), “El cliente {n[0]} aparece {n[1]} vez” o “El cliente {n[0]} aparece {n[1]} veces repetido”. Segundo caso (2), “El cliente {n[0]} tiene {n[1]} repetición seguida” o “El cliente {n[0]} tiene {n[1]} repeticiones seguidas”. Si el usuario ingresa una opción no valida. Debe retornar el mensaje: “Ingresa un número valido”



Ejemplos:

Primera Función:

ListaClientes	Return
["Santiago", "Santiago", "Santiago", "Laura", "Laura", "Laura", "Laura", "Pérez"]	<u>[('Santiago', 3), ('Laura', 4), ('Pérez', 1)]</u>
["Santiago", "Santiago", "Santiago", "Laura", "Laura", "Laura", "Laura", "Pérez", "Pérez", "Pérez", "Pérez", "Santiago", "Pedro"]	[('Santiago', 4), ('Laura', 4), ('Pérez', 3), ('Pedro', 1)]

Entrada:

Nombre	Tipo	Descripción
ListaClientes	list	Lista ,en donde se encuentra las personas registradas en una semana.

Salida:

Tipo de retorno	Descripción
List	[(cliente, Nveces)]. Una lista con las tuplas creadas, en donde se encuentra el nombre y después el número de veces repetidas

Esqueleto:

```
def VerCantidadRepetidos(ListaClientes: list) -> list:
```



Segunda Función:

ListaClientes	Return
["Santiago", "Santiago", "Santiago", "Laura", "Laura", "Laura", "Laura", "Pérez"]	<u>[('Santiago', 3), ('Laura', 4)]</u>
["Santiago", "Santiago", "Santiago", "Laura", "Laura", "Laura", "Laura", "Pérez", "Pérez", "Pérez", "Santiago", "Pedro"]	[('Santiago', 3), ('Laura', 4), ('Pérez', 3)]

Entrada:

Nombre	Tipo	Descripción
ListaClientes	list	Lista, en donde se encuentra las personas registradas en una semana.

Salida:

Tipo de retorno	Descripción
dict	[('cliente', Nveces)]. Una lista con las tuplas creadas, en donde se encuentra el nombre y después el número de veces repetidas de seguido

Esqueleto:

```
def VerRepetidosEnMismaLinea(ListaClientes: list) -> list:
```



Tercera Función:

ListaClientes	Opción	Return
["Santiago","Santiago","Santiago", "Laura", "Laura", "Laura", "Laura","Pérez"]	1	El cliente Santiago aparece 3 veces repetido El cliente Laura aparece 4 veces repetido El cliente Pérez aparece 1 vez
["Santiago","Santiago","Santiago", "Laura", "Laura", "Laura","Laura","Pérez","Pérez","Pérez","Santiago","Pedro"]	1	El cliente Santiago aparece 3 veces repetido El cliente Laura aparece 4 veces repetido El cliente Pérez aparece 3 veces repetido El cliente Pedro aparece 1 vez
["Santiago","Santiago","Santiago", "Laura", "Laura", "Laura", "Laura","Pérez"]	2	El cliente Santiago tiene 3 repeticiones seguidas El cliente Laura tiene 4 repeticiones seguidas
["Santiago","Santiago","Santiago", "Laura", "Laura", "Laura","Laura","Pérez","Pérez","Pérez","Santiago","Pedro"]	2	El cliente Santiago tiene 3 repeticiones seguidas El cliente Laura tiene 4 repeticiones seguidas El cliente Pérez tiene 3 repeticiones seguidas

Entrada:

Nombre	Tipo	Descripción
ListaClientes	list	Lista, en donde se encuentra las personas registradas en una semana.
Opción	Int	El número representa si queremos ver las veces repetidas (1) o si queremos ver las veces repetidas seguidas de un cliente (2)

Salida:

Tipo de retorno	Descripción
str	"El cliente {n[0]} aparece {n[1]} vez" o "El cliente {n[0]} aparece {n[1]} veces repetido". Segundo caso (2), "El cliente {n[0]} tiene {n[1]} repetición seguida" o "El cliente {n[0]} tiene {n[1]} repeticiones seguidas".

Esqueleto:

```
def ImprimirVectores(ListaClientes: list, Opcion: int) ->str:
```