

# Laboratorio 3

## Punto #1

1. De que forma podemos agregar, eliminar, modificar y recorrer una estructura tipo list?

### **Agregar:**

para agregar un nuevo elemento al final se utiliza el metodo `append("")`

ejem:

```
carros = ["BMW", "Aston Martin", "Chevrolet"] carros.append("Ferrari")
```

para agregar varios elementos al final de la lista, se utiliza el metodo `extend()`

ejem:

```
carros = ["BMW", "Aston Martin", "Chevrolet"] carros.extend(["Ferrari", "Fiat"])
```

agregar un elemento en una posicion determinada, con el metodo `insert(posicion, "nuevo elemento")`

ejem:

```
carros = ["BMW", "Aston Martin", "Chevrolet"] carros.insert(0, "Fiat")
```

### **Eliminar:**

para eliminar un elemento especifico en la lista se utiliza el metodo `remove()` o si conocemos la posicion se utiliza el metodo `pop()` o la palabra clave **del**

ejm

```
carros = ["BMW", "Aston Martin", "Chevrolet"] carros.remove("BMW")
```

```
carros = ["BMW", "Aston Martin", "Chevrolet"] carros.pop(1) // si no se conoce el numero de index pop() elimina el ultimo
```

```
carros = ["BMW", "Aston Martin", "Chevrolet"] del carros[1]
```

## Modificar

para modificar o cambiar un valor de la lista se puede utilizar con el numero de posicion donde se va modificar el dato

```
ejm
carros = ["BMW", "Aston Martin", "Chevrolet"]
carros[3] = "Ferrari" # Chevrolet se cambiaria por Ferrari
```

para modificarr varios valores en un cierto rango, se establecen el rango con los numeros de indexacion de la lista, si se ingresan mas datos que los indicados en el rango, se colocaria despues.

```
ejm
carros = ["BMW", "Aston Martin", "Chevrolet", "Ferrari", "Opel"]
carros[1:3] = ["Toyota", "Nissan",]
```

tambien se puede utilizar el metodo insert()

```
ejm
carros = ["BMW", "Aston Martin", "Chevrolet", "Ferrari", "Opel"]
carros.insert(2, "Nissan")
```

## Recorrer la lista

Para recorrer la lista se pude utilizar un loop for

```
ejem:
carros = ["BMW", "Aston Martin", "Chevrolet", "Ferrari", "Opel"]
for x in carros:
    if "B" in x:
        nuve_lista.append(x)

print(nuvea_lista)
```

Para recorrer la lista se pude utilizar un loop while

```

carros = ["BMW", "Aston Martin", "Chevrolet", "Ferrari", "Opel"]
i = 0
while i < len(carros):
    print(carros[i])
    i = i + 1

```

Utilizando comprension de lista

Con comprension de lista se acorta la sintaxis del loop

ejm

```

carros = ["BMW", "Aston Martin", "Chevrolet", "Ferrari", "Opel"]
[print(x) for x in carros]

```

```

carros = ["BMW", "Aston Martin", "Chevrolet", "Ferrari", "Opel"]
lista1 = []

```

2. De que forma podemos contar la cantidad de elementos que se encuentran en una colección tipo <list>

Para contar se utiliza el metodo count() si se quiere saber la cantidad de veces que esta un dato o el metodo len() para conocer el total de datos en la lista

3. Que entiende por indexación de los elementos en una colección de datos.

Entiendo que es el orden en como estan almacenados los datos y de que manera se pueden acceder o filtrar.

Es la organización de los datos y como estas son leidas o se tiene acceso, sirve para acceder de una forma rapida a la informacion ahorrando tiempo dinero mejorando la organización.