

Punto #1 (teórico) – [2.5%]

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

Para añadir un nuevo elemento a una lista se utiliza el método `append()` y para añadir varios elementos, el método `extend()`:

```
>>> vocales = ['a']
>>> vocales.append('e') # Añade un elemento
>>> vocales
['a', 'e']

>>> vocales.extend(['i', 'o', 'u']) # Añade un grupo de elementos
>>> vocales
['a', 'e', 'i', 'o', 'u']
```

Es posible modificar un elemento de una lista en Python con el operador de asignación `=`. Para ello, lo único que necesitas conocer es el índice del elemento que quieres modificar o el rango de índices:

```
>>> vocales = ['o', 'o', 'o', 'o', 'u']

# Actualiza el elemento del índice 0
>>> vocales[0] = 'a'
>>> vocales
['a', 'o', 'o', 'o', 'u']

# Actualiza los elementos entre las posiciones 1 y 2
>>> vocales[1:3] = ['e', 'i']
>>> vocales
['a', 'e', 'i', 'o', 'u']
```

Con la sentencia `del` se puede eliminar un elemento a partir de su índice:

```
# Elimina el elemento del índice 1
>>> vocales = ['a', 'e', 'i', 'o', 'u']
>>> del vocales[1]
>>> vocales
['a', 'i', 'o', 'u']

# Elimina los elementos con índices 2 y 3
>>> vocales = ['a', 'e', 'i', 'o', 'u']
>>> del vocales[2:4]
>>> vocales
['a', 'e', 'u']

# Elimina todos los elementos
>>> del vocales[:]
>>> vocales
[]
```

Ya hemos visto que se puede usar el bucle for en Python para recorrer los elementos de una secuencia. En nuestro caso, para recorrer una lista en Python utilizaríamos la siguiente estructura:

```
>>> colores = ['azul', 'blanco', 'negro']
>>> for color in colores:
    print(color)

azul
blanco
negro
```

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

Para acceder a un elemento de una lista se utilizan los índices. Un índice es un número entero que indica la posición de un elemento en una lista. El primer elemento de una lista siempre comienza en el índice 0.

Por ejemplo, en una lista con 4 elementos, los índices de cada uno de los ítems serían 0, 1, 2 y 3.

```
>>> lista = ['a', 'b', 'd', 'i', 'j']
>>> lista[0] # Primer elemento de la lista. Índice 0
'a'
>>> lista[3] # Cuarto elemento de la lista. Índice 3
'i'
```

3. ¿Qué entiende por indexación de los elementos en una colección de datos?

Para mí entender significa que se actualiza con datos nuevos, después de ser creada, o sea con el tiempo se va incluyendo datos o desapareciendo.