

# Práctica de Listas: Nivel 2

September 23, 2025

Este colab fue desarrollado por Arnold Charry Armero.

## 1 Práctica de Listas: Nivel 2

### 1.1 Ejercicio 1

Cree una función que le pregunte al usuario cuantos números desea que tenga una lista (por ejemplo 50), y con base en ese número, llene una lista por medio de números aleatorios (utilizando randint(0,101)) de manera que cada número aleatorio será cada valor de su lista. Como resultado usted debe retornar la lista llena de números.

```
[3]: def numeros():
    n = int(input("¿Cuántos números desea que se generen aleatoriamente?: "))
    return n

from random import randint

Milista = []

for i in range(numeros()):
    number = randint(0, 101)
    Milista.append(number)
print(f"La lista llena de números es {Milista}")
```

```
¿Cuántos números desea que se generen aleatoriamente?: 10
La lista llena de números es [25, 8, 21, 11, 90, 96, 73, 23, 21, 17]
```

### 1.2 Ejercicio 2

Realice un programa que reciba como parámetro una Lista con números enteros (si desea puede usar el punto 1 para generar su lista) y un número, de manera que la función genere una nueva lista que contenga sólo los números que son divisibles por el número ingresado por el usuario. Ejemplo: Si se tiene la lista [6, 7, 8, 10, 12, 15, 3, 9, 22, 20] y el número ingresado por el usuario es 4, se deberá generar la lista [8, 12, 20]

```
[4]: Milista = [6, 7, 8, 10, 12, 15, 3, 9, 22, 20]

def ListaDivisible(lista):
    Listadivisibles = []
```

```

número = int(input("Ingrese el número por el cual serán divisibles los elementos de la lista: "))

for i in lista:
    if i % número == 0:
        Listadivisibles.append(i)
return Listadivisibles

print(f"La lista de números divisibles es {ListaDivisible(Milista)}")

```

Ingrese el número por el cual serán divisibles los elementos de la lista: 4  
La lista de números divisibles es [8, 12, 20]

### 1.3 Ejercicio 3

Diseñe una función que tome los elementos que se encuentran en las posiciones impares de una lista, los anexe a otra lista y la retorne. Ejemplo: Si se tiene la lista [1, 2, 4, 5, 6, 4, 3, 2, 1] se debe retornar la lista [2, 5, 4, 2].

```

[6]: Milista = [1, 2, 4, 5, 6, 4, 3, 2, 1]

def ListaImpar(lista):
    Odd_list = []
    for i in range(len(lista)):
        if i % 2 != 0:
            Odd_list.append(lista[i])
    return Odd_list

print(f"La lista con los índices impares es {ListaImpar(Milista)}")

```

La lista con los índices impares es [2, 5, 4, 2]

### 1.4 Ejercicio 4

Diseñe un procedimiento en Python al cual le ingrese una lista de números enteros positivos. La función debe crear e imprimir una lista con los elementos pares y una lista con los elementos impares. Ejemplo: Si se tiene la lista [6, 7, 8, 10, 12, 15, 3, 9, 22, 20] se debe imprimir una lista con [6, 8, 10, 12, 22, 20] y otra lista con [7, 15, 3, 9]

```

[7]: Milista = [6, 7, 8, 10, 12, 15, 3, 9, 22, 20]

def ListasParesImpares(lista):
    ListaPar = []
    ListaImpar = []

    for i in lista:
        if i % 2 == 0:
            ListaPar.append(i)
        else:

```

```

    ListaImpar.append(i)

    print(f"Lista Par: {ListaPar}")
    print(f"Lista Impar: {ListaImpar}")

ListasParesImpares(Milista)

```

Lista Par: [6, 8, 10, 12, 22, 20]  
 Lista Impar: [7, 15, 3, 9]

## 1.5 Ejercicio 5

Realice un programa que se encargue de eliminar los espacios vacíos de una lista de caracteres (si los hay) al comienzo y final de la lista. Ejemplo. Si se tiene la lista [‘,’a’,’b’,’e’,’,’y’,’t’,’’] deberá quedar [‘a’,’b’,’e’,’,’y’,’t’].

```
[10]: Milista = ['', 'a', 'b', 'e', '', 'y', 't', '']

def EliminarEspacios(lista):
    for i in lista:
        if i == "":
            lista.remove(i)
    print(f"La lista sin espacios vacíos es {lista}")

EliminarEspacios(Milista)
```

La lista sin espacios vacíos es ['a', 'b', 'e', 'y', 't']

## 1.6 Ejercicio 6

Realice ahora un programa que se encargue de eliminar todos los espacios vacíos que se encuentren en una lista de caracteres.

```
[14]: Milista = ['', 'a', ' ', 'b', ' ', ' ', 'e', ' ', 'y', 't', '']

nueva_lista = []

for i in Milista:
    if i.strip() != '':
        nueva_lista.append(i)

print(nueva_lista)
```

['a', 'b', 'e', 'y', 't']

## 1.7 Ejercicio 7

Realice una función que recibe como parámetro dos listas. La primera lista LE contiene los elementos a reorganizar, la segunda lista LP contiene las nuevas posiciones de los elemento de la lista

LE. La función con los parámetros: LE = ["a", "b", "c", "d"] LP = [1, 3, 0, 2] debería retornar la lista ["c", "a", "d", "b"].

```
[17]: LE = ["a", "b", "c", "d"]
LP = [1, 3, 0, 2]
```

```
def Organizarlistas(lista, posiciones):
    nueva_lista = [None] * len(lista)
    for i in range(len(lista)):
        nueva_lista[posiciones[i]] = lista[i]
    print(f"La fila organizada es {nueva_lista}")
```

```
Organizarlistas(LE, LP)
```

```
La fila organizada es ['c', 'a', 'd', 'b']
```

## 1.8 Ejercicio 8

Realice un programa que basado en 2 listas, construya la intersección de conjuntos de ellas (los elementos que están en la lista A y los elementos que están en la lista B) en una tercera lista, la nueva lista NO PUEDE tener elementos repetidos.

```
[20]: lista1 = ["A", "B", "C", "D", "E"]
lista2 = ["B", "D", "F", "E", "H"]
```

```
nueva_lista = []

for i in lista1:
    for j in lista2:
        if i == j and i not in nueva_lista:
            nueva_lista.append(i)

print(f"La intersección de listas es {nueva_lista}")
```

```
La intersección de listas es ['B', 'D', 'E']
```

## 1.9 Ejercicio 9

Realice un programa basado en 2 listas, que construya la diferencia de conjuntos de ellas (los elementos que se encuentran en la lista A y que no se encuentran en la lista B) en una tercera lista.

```
[26]: lista1 = ["A", "B", "C", "D", "E"]
lista2 = ["B", "D", "F", "E", "H"]
```

```
nueva_lista = []

for i in lista1:
    if i not in lista2:
        nueva_lista.append(i)
```

```
print(f"La diferencia de conjuntos de listas es {nueva_lista}")
```

La diferencia de conjuntos de listas es ['A', 'C']

## 1.10 Ejercicio 10

Realice un programa que permita ingresar una cantidad N de artículos y una cantidad M de precios en el mercado, con esos datos cree una lista similar a la siguiente: Articulos=[[“Jabon”,2000,1800,1300], [“arroz”,1350,1500,1600.....X].....[X]]

```
[31]: Artículos = []

N = int(input("Ingresa cuántos artículos quieres agregar: "))
M = int(input("Ingresa cuántos precios quieres agregar por producto: "))

for i in range(N):
    Artículos.append([input("¿Cuál artículo quieres agregar?: ")])
    for j in range(M):
        Artículos[i].append(float(input("¿Cuáles son los precios por agregar?: ")))

print(f"La lista con Artículos es {Artículos}")
```

```
Ingresa cuántos artículos quieres agregar: 3
Ingresa cuántos precios quieres agregar por producto: 3
¿Cuál artículo quieres agregar?: Jabon
¿Cuáles son los precios por agregar?: 2000
¿Cuáles son los precios por agregar?: 1800
¿Cuáles son los precios por agregar?: 1300
¿Cuál artículo quieres agregar?: arroz
¿Cuáles son los precios por agregar?: 1350
¿Cuáles son los precios por agregar?: 1500
¿Cuáles son los precios por agregar?: 1600
¿Cuál artículo quieres agregar?: Maracuyá
¿Cuáles son los precios por agregar?: 2000
¿Cuáles son los precios por agregar?: 2300
¿Cuáles son los precios por agregar?: 2500
La lista con Artículos es [['Jabon', 2000.0, 1800.0, 1300.0], ['arroz', 1350.0, 1500.0, 1600.0], ['Maracuyá', 2000.0, 2300.0, 2500.0]]
```