



# ESCUELA POLITÉCNICA NACIONAL

## ESCUELA DE FORMACIÓN DE TECNÓLOGOS



### ALGORITMOS Y ESTRUCTURAS DE DATOS (TSDS)

ASIGNATURA:

ALGORITMOS Y ESTRUCTURAS DE DATOS

PROFESOR:

Ing. Lorena Chulde MSc.

PERÍODO ACADÉMICO:

2023-B

### TAREA Grupal

### TÍTULO:

ARREGLOS – PARÁMETROS POR REFERENCIA Y VALOR - MATRICES

**Nombres de los estudiantes:**

Incluir el nombre de todos los estudiantes del grupo



2023-B

## PROPÓSITO DE LA TAREA

Reutilizar el código mediante funciones para una programación óptima.

## OBJETIVO

Facilitar la modularidad y reutilización de código, permitiendo a los programadores dividir programas en bloques más pequeños y manejables, tomando en cuenta que, al utilizar funciones, podemos reutilizar código existente en diferentes partes de un programa, evitando así la repetición innecesaria de líneas de código. Además, las funciones nos permiten encapsular lógica específica en un solo lugar, lo que facilita la depuración y el mantenimiento del código a largo plazo.

## Parte I

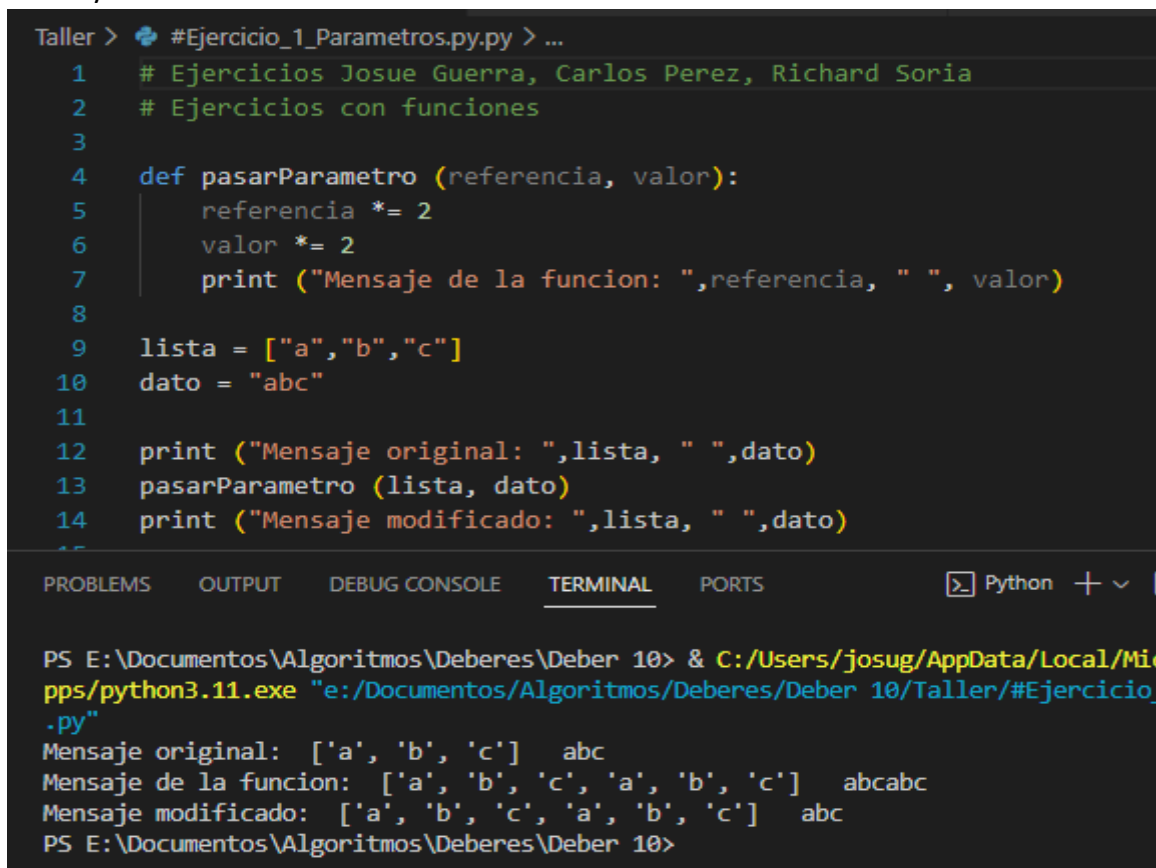
### TALLER:

## Listas – Funciones

Leer la guía sobre Paso de parámetros por valor y referencia del siguiente link:

[Paso de parámetros por lista](#)

1. Crear una función con dos parámetros paso por referencia y valor, que doble el valor de la lista y el dato enviado



```
Taller > #Ejercicio_1_Parametros.py.py > ...
1  # Ejercicios Josue Guerra, Carlos Perez, Richard Soria
2  # Ejercicios con funciones
3
4  def pasarParametro (referencia, valor):
5      referencia *= 2
6      valor *= 2
7      print ("Mensaje de la funcion: ",referencia, " ", valor)
8
9  lista = ["a","b","c"]
10 dato = "abc"
11
12 print ("Mensaje original: ",lista, " ",dato)
13 pasarParametro (lista, dato)
14 print ("Mensaje modificado: ",lista, " ",dato)
15

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Python + v

PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/Local/Microsoft/WindowsApps/python3.11.exe "e:/Documentos/Algoritmos/Deberes/Deber 10/Taller/#Ejercicio_1_Parametros.py.py"
Mensaje original: ['a', 'b', 'c']  abc
Mensaje de la funcion: ['a', 'b', 'c', 'a', 'b', 'c']  abcabcd
Mensaje modificado: ['a', 'b', 'c', 'a', 'b', 'c']  abc
PS E:\Documentos\Algoritmos\Deberes\Deber 10>
```

## 2. Crear una función que muestre una lista dada

```
Taller > #Ejercicio_2_Funcion-lista.py.py > ...
1  # Ejercicios Josue Guerra, Carlos Perez, Richard Soria
2  # Funcion Mostrar una lista
3
4  def mostrarLista (lista):
5      for elemento in lista:
6          return lista
7
8  lista = [5,6,7,8,9]
9  print (mostrarLista (lista))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/
pps/python3.11.exe "e:/Documentos/Algoritmos/Deberes/Deber 10/Taller/#E
.py.py"
[5, 6, 7, 8, 9]
PS E:\Documentos\Algoritmos\Deberes\Deber 10>
```

## 3. Crear una función que sume los valores de una lista dada

```
Taller > #Ejercicio_3_Funcion-suma.py > ...
1  # Ejercicios Carlos Perez, JOsue Guerra, Richar Soria
2  # Suma elemntos de la lista
3
4  def sumarElementos (lista):
5      contador = 0
6
7      for elemento in lista:
8          contador += elemento
9      return contador
10
11 lista = [9,8,7,6]
12 print(lista)
13 print (sumarElementos (lista))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python

```
PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/
pps/python3.11.exe "e:/Documentos/Algoritmos/Deberes/Deber 10/Taller/#E
py"
[9, 8, 7, 6]
30
PS E:\Documentos\Algoritmos\Deberes\Deber 10>
```

4. Elaborar tres funciones, la primera recibe la lista y retorna la suma de todos sus elementos, la segunda recibe la lista y retorna el mayor valor y la última recibe la lista y retorna el menor.

```
C: > Users > josug > Downloads > #Ejercicio5_Parametros_GuerraJosue.py > ...
1  # Tarea Carlos Perez, Josue Guerra, Richard Soria
2  #Crear 3 funciones: sumar, encontrarMax, encontrarMin
3  def sumar(lista):
4      suma = 0
5      for elemento in lista:
6          suma = suma + elemento
7      return suma
8  def encontrarMax(lista):
9      maximo = lista[0]
10     for elemento in lista:
11         if elemento > maximo:
12             maximo = elemento
13     return maximo
14 def encontrarMin(lista):
15     minimo = lista[0]
16     for elemento in lista:
17         if elemento < minimo:
18             minimo = elemento
19     return minimo
20 cantidad = int(input("Ingrese la cantidad de elementos en la lista: "))
21 lista = []
22 for i in range(cantidad):
23     valor = int(input("Ingrese el valor del elemento: "))
24     lista.append(valor)
25
26 print(lista)
27 print("La suma de los elementos de la lista es:", sumar(lista))
28 print("El mayor elemento de la lista es:", encontrarMax(lista))
29 print("El menor elemento de la lista es:", encontrarMin(lista))
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS C:\Users\josug> & C:/Users/josug/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:\Users\josug\Downloads\#Ejercicio5_Parametros_GuerraJosue.py
Ingrese la cantidad de elementos en la lista: 5
Ingrese el valor del elemento: 1
Ingrese el valor del elemento: 5
Ingrese el valor del elemento: 9
Ingrese el valor del elemento: 6
Ingrese el valor del elemento: 3
[1, 5, 9, 6, 3]
La suma de los elementos de la lista es: 24
El mayor elemento de la lista es: 9
El menor elemento de la lista es: 1
```

## Algoritmos de búsqueda lineal

## 5. Crear una función que imprima la posición de un valor enviado

`def buscarLinealmente(lista,numero):`

```
i = 0 //posicion de la lista
for elemento in lista: # estamos en la posicion i, z contiene el valor de lista[i]
    if elemento == numero:
        return i
    i = i+1
return -1
```

`lista = [2,5,3,6,7,8,4,9,1,0]`

`print(buscarLinealmente(lista, 7))`

```
Taller > #Ejercicio_5_Posicion-Funciones.py.py > ...
1 # Ejercicios Josue Guerra, Carlos Perez, Richard Soria
2 #Crear 3 funciones: sumar, encontrarMax, encontrarMin
3 def sumar(lista):
4     suma = 0
5     for elemento in lista:
6         suma = suma + elemento
7     return suma
8 def encontrarMax(lista):
9     maximo = lista[0]
10    for elemento in lista:
11        if elemento > maximo:
12            maximo = elemento
13    return maximo
14 def encontrarMin(lista):
15    minimo = lista[0]
16    for elemento in lista:
17        if elemento < minimo:
18            minimo = elemento
19    return minimo
20 cantidad = int(input("Ingrese la cantidad de elementos en la lista: "))
21 lista = []
22 for i in range(cantidad):
23     valor = int(input("Ingrese el valor del elemento: "))
24     lista.append(valor)
25
26 print(lista)
27 print("La suma de los elementos de la lista es:", sumar(lista))
28 print("El mayor elemento de la lista es:", encontrarMax(lista))
29 print("El menor elemento de la lista es:", encontrarMin(lista))
```

EJECUCIÓN:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/A
python3.11.exe "e:/Documentos/Algoritmos/Deberes/Deber 10/Taller
.py"
Ingrese la cantidad de elementos en la lista: 5
Ingrese el valor del elemento: 1
Ingrese el valor del elemento: 5
Ingrese el valor del elemento: 9
Ingrese el valor del elemento: 6
Ingrese el valor del elemento: 3
[1, 5, 9, 6, 3]
La suma de los elementos de la lista es: 24
El mayor elemento de la lista es: 9
El menor elemento de la lista es: 1
PS E:\Documentos\Algoritmos\Deberes\Deber 10> █
```

## Matrices

valores =  $\begin{bmatrix} 8, 14, -6 \\ 12, -7, 4 \\ -11, 3, 21 \end{bmatrix}$

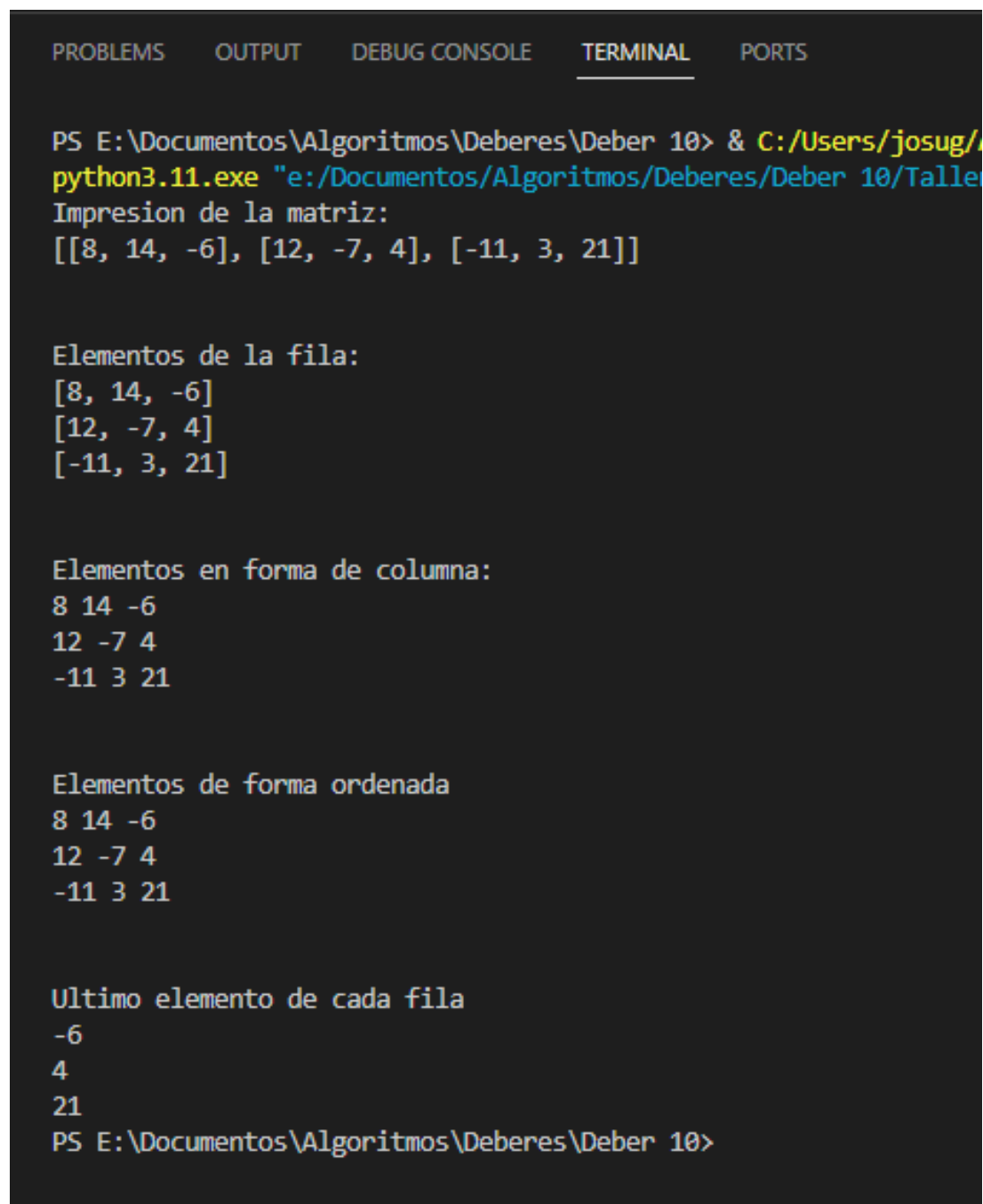
6. Dada la siguiente matriz:

- Mostrar todos los elementos de la matriz fila por fila
- Mostrar todos los elementos de la matriz elemento a elemento en columna
- Mostrar todos los elementos de la matriz en formato de matriz
- Leer el último elemento de cada fila

```
Taller > #Ejercicio_6_Matrices.py.py > ...
1  # Ejercicios Josue Guerra, Carlos Perez, Richard Soria
2  # Matrices
3
4  matriz = [[8,14,-6],[12,-7,4],[-11,3,21]]
5  print ("Impresion de la matriz: ")
6  print (matriz)
7  print ("\n")
8
9  # Mostrar todos los elementos de la matriz fila por fila
10
11 print ("Elementos de la fila: ")
12 for fila in matriz:
13     print (fila)
14 print ("\n")
15
16 # Mostrar todos los elementos de la matriz elemento a elemento en columna
17
18 print ("Elementos en forma de columna: ")
19 for row in matriz:
20     for elemento in row:
21         print(elemento, end=" ")
22     print()
23 print ("\n")
24
25 # Mostrar todos los elementos de la matriz en formato de matriz
26
27 print ("Elementos de forma ordenada")
28 for row in matriz:
29     for elemento in row:
30         print(elemento, end=" ")
31     print()
32 print ("\n")
33
34 # Leer el ultimo elemento de cada fila
35
36 print ("Ultimo elemento de cada fila")
37 long= len(matriz)
38 for e in range(long):
39     print(matriz[e][-1])
```

LA EJECUCIÓN DEL CÓDIGO SE ENCUENTRA EN LA SIGUIENTE PÁGINA

Ejecución:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/
python3.11.exe "e:/Documentos/Algoritmos/Deberes/Deber 10/Talle
Impresion de la matriz:
[[8, 14, -6], [12, -7, 4], [-11, 3, 21]]

Elementos de la fila:
[8, 14, -6]
[12, -7, 4]
[-11, 3, 21]

Elementos en forma de columna:
8 14 -6
12 -7 4
-11 3 21

Elementos de forma ordenada
8 14 -6
12 -7 4
-11 3 21

Ultimo elemento de cada fila
-6
4
21
PS E:\Documentos\Algoritmos\Deberes\Deber 10>
```

EL EJERCICIO 7 DEL TALLER SE ENCUENTRA EN LA SIGUIENTE PÁGINA

7. Generar una matriz con elementos de tipo entero desde el teclado. El usuario debe especificar las filas y las columnas de la matriz e ingresar los elementos, mostrar en formato de matriz

```
Taller > #Ejercicio_7_Matriz-Entero.py.py > ...
1  # Ejercicios Josue Guerra, Carlos Perez, Richard Soria
2  # Generar una matriz con elementos enteros desde el
3  # teclado que el usuario especifique el numero de
4  # filas y columnas y mostrar en formato matriz
5
6  matriz = []
7  filas = int (input ("Ingrese el numero de filas: "))
8  columnas = int (input ("Ingrese el numero de columnas: "))
9
10 for fila in range (filas):
11     fila = []
12     for elemento in range (columnas):
13         elemento = int (input ("Ingrese el elemento: "))
14         fila.append (elemento)
15     matriz.append (fila)
16
17 print ("La matriz generada es: ")
18 for fila in matriz:
19     for elemento in fila:
20         print(elemento, end=" ")
21     print()
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS

```
PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/L
python3.11.exe "e:/Documentos/Algoritmos/Deberes/Deber 10/Taller/#Ejerci
Ingrese el numero de filas: 2
Ingrese el numero de columnas: 3
Ingrese el elemento: 1
Ingrese el elemento: 5
Ingrese el elemento: 9
Ingrese el elemento: 6
Ingrese el elemento: 8
Ingrese el elemento: 7
La matriz generada es:
1 5 9
6 8 7
PS E:\Documentos\Algoritmos\Deberes\Deber 10> |
```



## Parte II

### TAREA:

#### Funciones

1. Crear una función llamada "iniciarSesion", que recibe un nombre de usuario y una contraseña y te devuelve Verdadero si el nombre de usuario es "usuario1" y la contraseña es "asdasd". Además, recibe el número de intentos que se ha intentado iniciar sesión y si no se ha podido iniciar la sesión incrementa este valor.

Crear un programa principal donde se pida un nombre de usuario y una contraseña y se intente hacer login, solamente tenemos tres oportunidades para intentarlo.

```
Deber > #Ejercicio_1_Iniciar-sesion.py.py > ...
1  # Ejercicios Tarea Josue Guerra, Carlos Perez, Richard Soria
2  # Crear una función llamada "iniciarSesion", que recibe
3  # un nombre de usuario y una contraseña si el nombre de usuario
4  # es "usuario1" y la contraseña es "asdasd". Además recibe el
5  # número de intentos que se ha intentado iniciar sesión
6
7  def iniciarSesion (usuario, contraseña, intentos):
8      if usuario == "usuario1" and contraseña == "asdasd":
9          return True
10     else:
11         intentos += 1
12         return False
13 intentos = 0
14
15 while intentos < 3:
16     usuario = input("Ingrese su usuario: ")
17     contraseña = input("Ingrese su contraseña: ")
18
19     if iniciarSesion(usuario, contraseña, intentos):
20         print("Inicio de sesión exitoso")
21         break
22     else:
23         intentos += 1
24         print("Nombre de usuario o contraseña incorrectos. Por favor, inténtelo de nuevo.")
25
26 if intentos == 3:
27     print("Se han agotado los tres intentos. Intente más tarde.")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/Local/Microsoft/WindowsApps/python3.11.exe r-sesion.py.py
Ingrese su usuario: Josue
Ingrese su contraseña: 15968
Nombre de usuario o contraseña incorrectos. Por favor, inténtelo de nuevo.
Ingrese su usuario: usuario1
Ingrese su contraseña: asdasd
Inicio de sesión exitoso
PS E:\Documentos\Algoritmos\Deberes\Deber 10> 
```

2. Crea una función “ConvertirEspaciado”, que reciba como parámetro un texto y devuelve una cadena con un espacio adicional tras cada letra. Por ejemplo, “Hola, tú” devolverá “H o l a , t ú “. Crea un programa principal donde se use dicha función.

```

Deber > #Ejercicio_2_Funcion-espacio.py.py > ...
1  # Ejercicios Tarea Josue Guerra, Carlos Perez, Richard Soria
2
3  # Crea un función “ConvertirEspaciado”, que reciba como
4  # parámetro un texto y devuelve una cadena con un espacio adicional tras cada letra.
5  # Por ejemplo, “Hola, tú” devolverá “H o l a , t ú “.
6  # Crea un programa principal donde se use dicha función.
7
8  def ConvertirEspaciado(texto):
9      resultado = ""
10     for letra in texto:
11         resultado += letra + " "
12     return resultado
13
14     texto = input("Ingrese un texto: ")
15     resultado = ConvertirEspaciado(texto)
16     print("El resultado es:", resultado)
17
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/Local/Microsoft/WindowsApps/python-espacio.py.py
Ingrese un texto: hola, que la fuerza te acompañe
El resultado es: h o l a ,   q u e   l a   f u e r z a   t e   a c o m p a ñ e
PS E:\Documentos\Algoritmos\Deberes\Deber 10>

```

3. Crear varias funciones con el fin de que el usuario ingrese por teclado la longitud de una lista, los valores de la lista e imprima: la lista, la posición y el valor indicado por el usuario. (Algoritmos de búsqueda lineal)

```

Deber > #Ejercicio_3_Funcion-valor.py.py > ...
1  # Ejercicios Tarea Josue Guerra, Carlos Perez, Richard Soria
2
3  # Crear varias funciones con el fin de que el usuario ingrese
4  # por teclado la longitud de una lista, los valores de la lista
5  # e imprima: la lista, la posición y el valor indicado por el usuario.
6  # (Algoritmos de búsqueda lineal)
7
8  def crearLista(longitud):
9      lista = []
10     for i in range(longitud):
11         valor = int(input("Ingrese un numero para la posicion {}: ".format(i)))
12         lista.append(valor)
13     return lista
14
15     def buscarValor(lista, valor):
16         for i in range(len(lista)):
17             if lista[i] == valor:
18                 return i
19         return -1
20
21     def imprimirResultado (lista, posicion, valor):
22         print("La lista es:", lista)
23         if posicion != -1:
24             print("El valor {} se encuentra en la posición {}".format(valor, posicion))
25         else:
26             print("El valor {} no se encuentra en la lista.".format(valor))
27
28     # Programa Principal
29
30     longitud = int(input("Ingrese la longitud de la lista: "))
31     lista = crearLista (longitud)
32     print ("La lista ingresada es: ",lista)
33
34     valor = int(input("Ingrese el valor que desea buscar en la lista: "))
35     posicion = buscarValor (lista, valor)
36
37     imprimirResultado (lista, posicion, valor)

```

Ejecución:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/
thon3.11.exe "e:/Documentos/Algoritmos/Deberes/Deber 10/D
Ingrese la longitud de la lista: 6
Ingrese un numero para la posicion 0: 1
Ingrese un numero para la posicion 1: 5
Ingrese un numero para la posicion 2: 6
Ingrese un numero para la posicion 3: 3
Ingrese un numero para la posicion 4: 8
Ingrese un numero para la posicion 5: 7
La lista ingresada es: [1, 5, 6, 3, 8, 7]
Ingrese el valor que desea buscar en la lista: 5
La lista es: [1, 5, 6, 3, 8, 7]
El valor 5 se encuentra en la posición 1.
PS E:\Documentos\Algoritmos\Deberes\Deber 10>

```

4. Imprimir una matriz cuyos valores sean ingresados por el usuario, además, se deberá ingresar las filas y las columnas.

```

Deber > #Ejercicio_4_Matriz-usuario.py.py > ...
1  # Ejercicios Tarea Josue Guerra, Carlos Perez, Richard Soria
2
3  # Imprimir una matriz cuyos valores sean ingresados por
4  # el usuario, además, se deberá ingresar las filas y las columnas.
5
6  filas = int(input("Ingrese el número de filas: "))
7  columnas = int(input("Ingrese el número de columnas: "))
8
9  matriz = []
10 for i in range(filas):
11     fila = []
12     for j in range(columnas):
13         valor = input("Ingrese el valor: ")
14         fila.append(valor)
15     matriz.append(fila)
16
17 print("La matriz ingresada es:")
18 for fila in matriz:
19     print(fila)

```

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Python
PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/Local/Micr
thon3.11.exe "e:/Documentos/Algoritmos/Deberes/Deber 10/Deber/#Ejercicio_4_Matriz
Ingrese el número de filas: 3
Ingrese el número de columnas: 3
Ingrese el valor: 1
Ingrese el valor: 5
Ingrese el valor: 6
Ingrese el valor: 8
Ingrese el valor: 9
Ingrese el valor: 7
Ingrese el valor: 1
Ingrese el valor: 2
Ingrese el valor: 6
La matriz ingresada es:
['1', '5', '6']
['8', '9', '7']
['1', '2', '6']
PS E:\Documentos\Algoritmos\Deberes\Deber 10>

```

5. Crear un programa para trabajar con una pila. Una pila es una estructura de datos que nos permite guardar un conjunto de variables. La característica fundamental es que el último elemento que se añade al conjunto es el primero que se puede sacar.  
Para representar una pila vamos a utilizar una lista de cadenas de caracteres.  
Vamos a crear varias funciones para trabajar con la pila:  
LongitudPila: Función que recibe una pila y devuelve el número de elementos que tiene.  
EstaVacíaPila: Función que recibe una pila y que devuelve si la pila está vacía, no tiene elementos.  
EstaLlenaPila: Función que recibe una pila y que devuelve si la pila está llena.  
AddPila: función que recibe una cadena de caracteres y una pila, y añade la cadena a la pila, si no está llena. Si está llena muestra un mensaje de error.  
SacarDeLaPila: Función que recibe una pila y devuelve el último elemento añadido y lo borra de la pila. Si la pila está vacía muestra un mensaje de error.  
EscribirPila: Función que recibe una pila y muestra en pantalla los elementos de la pila.
6. Realiza un programa principal que nos permita usar las funciones anteriores, que nos muestre un menú, con las siguientes opciones:  
Añadir elemento a la pila  
Sacar elemento de la pila  
Longitud de la pila  
Mostrar pila  
Salir

```
Deber > #Ejercicio_5_Funcion-pila.py.py > ...
1  # Ejercicios Tarea Josue Guerra, Carlos Perez, Richard Soria
2  def LongitudPila(pila):
3      return len(pila)
4
5  def EstaVacíaPila(pila):
6      return len(pila) == 0
7
8  def EstaLlenaPila(pila, tamaño_maximo):
9      return len(pila) == tamaño_maximo
10
11 def AddPila(cadena, pila, tamaño_maximo):
12     if not EstaLlenaPila(pila, tamaño_maximo):
13         pila.append(cadena)
14     else:
15         print("La pila está llena. No se puede añadir más elementos.")
16
17 def SacarDeLaPila(pila):
18     if not EstaVacíaPila(pila):
19         return pila.pop()
20     else:
21         print("La pila está vacía. No se puede sacar ningún elemento.")
22
23 def EscribirPila(pila):
24     print("Los elementos de la pila son:")
25     for elemento in pila:
26         print(elemento)
27
```

```
28 # Programa Principal
29 # Realiza un programa que permita
30 # usar las funciones anterior, que
31 # muestre un menú, con las siguientes opciones:
32 # Añadir elemento a la pila
33 # Sacar elemento de la pila
34 # Longitud de la pila
35 # Mostrar pila
36 # Salir
37
38 tamano_maximo = int (input ("Ingrese el tamaño maximo de la pila: "))
39 pila = []
40
41 op = 0
42 while op != 5:
43     print("Menú:")
44     print("1. Añadir elemento a la pila")
45     print("2. Sacar elemento de la pila")
46     print("3. Longitud de la pila")
47     print("4. Mostrar pila")
48     print("5. Salir")
49
50     opcion = input("Elige una opción: ")
51
52     if opcion == "1":
53         cadena = input ("Introduce una cadena: ")
54         AddPila(cadena, pila, tamano_maximo)
55
56     elif opcion == "2":
57         elemento = SacarDeLaPila(pila)
58         if elemento is not None:
59             print("Elemento sacado de la pila:", elemento)
60
61     elif opcion == "3":
62         print("Longitud de la pila:", LongitudPila(pila))
63
64     elif opcion == "4":
65         EscribirPila(pila)
66
67     elif opcion == "5":
68         print ("Gracias por usar nuestro programa")
69         exit (0)
```

El programa toma en consideración la orden del ejercicio 5 y la del ejercicio, juntando todo en un programa el cual se presenta en el siguiente código, por ende, se muestran en las capturas presentadas, su ejecución se encuentra en la siguiente página.

Ejecución:

Al ejecutar el código se comprueba que todas las opciones cumplan con lo solicitado, entonces se procede a escoger cada una de las opciones para visualizar en pantalla

```
PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/Local/Programs/Python/Python3.11.exe "e:/Documentos/Algoritmos/Deberes/Deber 10/Deber/#Ejercicio1.py"
Ingrese el tamaño máximo de la pila: 2
Menú:
1. Añadir elemento a la pila
2. Sacar elemento de la pila
3. Longitud de la pila
4. Mostrar pila
5. Salir
Elige una opción: 1
Introduce una cadena: 15
Menú:
1. Añadir elemento a la pila
2. Sacar elemento de la pila
3. Longitud de la pila
4. Mostrar pila
5. Salir
Elige una opción: 1
Introduce una cadena: 2
Menú:
1. Añadir elemento a la pila
2. Sacar elemento de la pila
3. Longitud de la pila
4. Mostrar pila
5. Salir
Elige una opción: 1
Introduce una cadena: 5
La pila está llena. No se puede añadir más elementos.
Menú:
1. Añadir elemento a la pila
2. Sacar elemento de la pila
3. Longitud de la pila
4. Mostrar pila
5. Salir
Elige una opción: 4
Los elementos de la pila son:
15
2
```

```
Menú:  
1. Añadir elemento a la pila  
2. Sacar elemento de la pila  
3. Longitud de la pila  
4. Mostrar pila  
5. Salir
```

```
Elige una opción: 2
```

```
Elemento sacado de la pila: 2
```

```
Menú:  
1. Añadir elemento a la pila  
2. Sacar elemento de la pila  
3. Longitud de la pila  
4. Mostrar pila  
5. Salir
```

```
Elige una opción: 4
```

```
Los elementos de la pila son:  
15
```

```
Menú:  
1. Añadir elemento a la pila  
2. Sacar elemento de la pila  
3. Longitud de la pila  
4. Mostrar pila  
5. Salir
```

```
Elige una opción: 1
```

```
Introduce una cadena: 8
```

```
Menú:  
1. Añadir elemento a la pila  
2. Sacar elemento de la pila  
3. Longitud de la pila  
4. Mostrar pila  
5. Salir
```

```
Elige una opción: 4
```

```
Los elementos de la pila son:  
15
```

```
8
```

```
Menú:  
1. Añadir elemento a la pila  
2. Sacar elemento de la pila  
3. Longitud de la pila  
4. Mostrar pila  
5. Salir
```

```
Elige una opción: 5
```

```
Gracias por usar nuestro programa
```

```
PS E:\Documentos\Algoritmos\Deberes\Deber 10> █
```

## 7. Crear una matriz dada por el usuario y mostrar la diagonal principal de una matriz

```

Deber > #Ejercicio_7_Diagonal-matriz.py > ...
1  # Ejercicios Tarea Josue Guerra, Carlos Perez, Richard Soria
2  # Crear una matriz dada por el usuario y mostrar la diagonal principal de una matriz
3
4  # Pide el número de filas y columnas de la matriz
5  while (True):
6      filas = int(input("Introduzca el número de filas de la matriz: "))
7      columnas = int(input("Introduzca el número de columnas de la matriz: "))
8      if (filas>0 and columnas>0):
9          break
10     else:
11         print("Ingrese un número de filas y columnas válidas.")
12
13  matriz = []
14
15  # Llenar matriz con datos digitados por el usuario
16  for fila in range(filas):
17      fila_arreglo = []
18      for columna in range(columnas):
19          fila_arreglo.append(int(input("Ingrese el número de la posición [" + str(fila) + "][" + str(columna) + "]: ")))
20      matriz.append(fila_arreglo)
21
22  print("MATRIZ ORIGINAL INGRESADA")
23
24  for i in matriz:
25      for j in i:
26          print(j, end=" ")
27      print()
28
29  print("DIAGONAL DE LA MATRIZ PRINCIPAL ")
30
31  for i in range(len(matriz)):
32      for j in range(len(matriz[i])):
33          if (i == j):
34              print(matriz[i][j], end=" ")
35          else:
36              print(" ", end=" ")
37      print()
38

```

Ejecución:

```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/Local/Programs/Python/Python311/python3.11.exe "e:/Documentos/Algoritmos/Deberes/Deber 10/Deber/Deber_7_Diagonal-matriz.py"

Introduzca el número de filas de la matriz: 4
Introduzca el número de columnas de la matriz: 4
Ingrese el número de la posición [0][0]: 1
Ingrese el número de la posición [0][1]: 5
Ingrese el número de la posición [0][2]: 9
Ingrese el número de la posición [0][3]: 6
Ingrese el número de la posición [1][0]: 2
Ingrese el número de la posición [1][1]: 4
Ingrese el número de la posición [1][2]: 3
Ingrese el número de la posición [1][3]: 1
Ingrese el número de la posición [2][0]: 8
Ingrese el número de la posición [2][1]: 7
Ingrese el número de la posición [2][2]: 2
Ingrese el número de la posición [2][3]: 0
Ingrese el número de la posición [3][0]: 6
Ingrese el número de la posición [3][1]: 8
Ingrese el número de la posición [3][2]: 2
Ingrese el número de la posición [3][3]: 1
MATRIZ ORIGINAL INGRESADA
1 5 9 6
2 4 3 1
8 7 2 0
6 8 2 1
DIAGONAL DE LA MATRIZ PRINCIPAL
1
4
2
1
PS E:\Documentos\Algoritmos\Deberes\Deber 10>
PS E:\Documentos\Algoritmos\Deberes\Deber 10>

```



## 8. Copiar una matriz a otra matriz

```
Deber > #Ejercicio_8_copia-matriz.py > ...
1  # Ejercicios Tarea Josue Guerra, Carlos Perez, Richard Soria
2  # Copiar una matriz a otra matriz
3
4  # Pide el número de filas y columnas de la matriz
5  while (True):
6      filas = int(input("Introduzca el número de filas de la matriz: "))
7      columnas = int(input("Introduzca el número de columnas de la matriz: "))
8      if (filas>0 and columnas>0):
9          break
10     else:
11         print("Ingrese un número de filas y columnas válidas.")
12
13  matriz = []
14  # Matriz original
15  for fila in range(filas):
16      fila_arreglo = []
17      for columna in range(columnas):
18          fila_arreglo.append(int(input("Ingrese el número de la posición [" + str(fila) + "][" + str(columna) + "]: ")))
19      matriz.append(fila_arreglo)
20
21  # Matriz copia
22  matriz_copia = []
23  for fila in range(filas):
24      fila_arreglo = []
25      for columna in range(columnas):
26          fila_arreglo.append(0)
27      matriz_copia.append(fila_arreglo)
28
29  print("MATRIZ ORIGINAL INGRESADA")
30  for i in matriz:
31      for j in i:
32          print(j, end=" ")
33      print()
34
35  print("MATRIZ VACÍA A COPIAR")
36  for i in matriz_copia:
37      for j in i:
38          print(j, end=" ")
39      print()
40
41  # Limpiar la matriz
42  matriz_copia.clear()
43  # Copiar los datos de la matriz original
44  for fila in range(filas):
45      fila_arreglo = []
46      for columna in range(columnas):
47          fila_arreglo.append(matriz[fila][columna])
48      matriz_copia.append(fila_arreglo)
49
50  print("MATRIZ COPIADA")
51  for i in matriz_copia:
52      for j in i:
53          print(j, end=" ")
54      print()
```

La ejecución del código se encuentra en la siguiente hoja.

Ejecución:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/A
s/python3.11.exe "e:/Documentos/Algoritmos/Deberes/Deber 10/Debe
..

Introduzca el número de filas de la matriz: 2
Introduzca el número de columnas de la matriz: 3
Ingresa el número de la posición [0][0]: 1
Ingresa el número de la posición [0][1]: 5
Ingresa el número de la posición [0][2]: 9
Ingresa el número de la posición [1][0]: 6
Ingresa el número de la posición [1][1]: 2
Ingresa el número de la posición [1][2]: 7
MATRIZ ORIGINAL INGRESADA
1 5 9
6 2 7
MATRIZ VACÍA A COPIAR
0 0 0
0 0 0
MATRIZ COPIADA
1 5 9
6 2 7
PS E:\Documentos\Algoritmos\Deberes\Deber 10>

```

## 9. Genere la matriz transpuesta de una matriz dada

```

Deber > #Ejercicio_9_Transponer-matriz.py > ...
1  # Ejercicios Tarea Josue Guerra, Carlos Perez, Richard Soria
2  # Genere la matriz transpuesta de una matriz dada
3
4  # Pide el número de filas y columnas de la matriz
5  while (True):
6      filas = int(input("Introduzca el número de filas de la matriz: "))
7      columnas = int(input("Introduzca el número de columnas de la matriz: "))
8      if (filas>0 and columnas>0):
9          break
10     else:
11         print("Ingresa un número de filas y columnas válidas.")
12
13     matriz = []
14
15     # Matriz original
16     for fila in range(filas):
17         fila_arreglo = []
18         for columna in range(columnas):
19             fila_arreglo.append(int(input("Ingresa el número de la posición [" + str(fila) + "][" + str(columna) + "]: ")))
20         matriz.append(fila_arreglo)
21
22     # Mostrar Matriz original
23     print("MATRIZ ORIGINAL INGRESADA")
24
25     for i in matriz:
26         for j in i:
27             print(j, end=" ")
28         print()
29
30     # Mostar Matriz transpuesta
31     print("MATRIZ TRASNPUESTA")
32
33     for i in range(len(matriz)):
34         for j in range(len(matriz[i])):
35             print(matriz[j][i], end=" ")
36         print()

```

## EJECUCIÓN

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/Local/Microsoft/WindowsApps/python3.11.exe "
oner-matriz.py.py"
Introduzca el número de filas de la matriz: 3
Introduzca el número de columnas de la matriz: 3
Ingresa el número de la posición [0][0]: 1
Ingresa el número de la posición [0][1]: 6
Ingresa el número de la posición [0][2]: 8
Ingresa el número de la posición [1][0]: 2
Ingresa el número de la posición [1][1]: 4
Ingresa el número de la posición [1][2]: 9
Ingresa el número de la posición [2][0]: 6
Ingresa el número de la posición [2][1]: 3
Ingresa el número de la posición [2][2]: 4
MATRIZ ORIGINAL INGRESADA
1 6 8
2 4 9
6 3 4
MATRIZ TRASNPUESTA
1 2 6
6 4 3
8 9 4
PS E:\Documentos\Algoritmos\Deberes\Deber 10>

```

## 10. Sume dos matrices

```

Deber > #Ejercicio_10_suma-matriz.py.py > ...
1  # Ejercicios Tarea Josue Guerra, Carlos Perez, Richard Soria
2  # Sume dos matrices
3
4  while (True):
5      filas = int(input("Introduzca el número de filas de las matrices: "))
6      columnas = int(input("Introduzca el número de columnas de la matrices: "))
7      if (filas>0 and columnas>0):
8          break
9      else:
10         print("Ingresa un número de filas y columnas válidas.")
11
12  matriz = []
13  print("Ingresa los elementos de la matriz 1")
14  # Llenar matriz 1 con datos digitados por el usuario
15  for fila in range(filas):
16      fila_arreglo = []
17      for columna in range(columnas):
18          fila_arreglo.append(int(input("Ingresa el número de la posición [" + str(fila) + "][" + str(columna) + "]: ")))
19      matriz.append(fila_arreglo)
20
21  matriz_2 = []
22  print("Ingresa los elementos de la matriz 2")
23  # Llenar matriz 2 con datos digitados por el usuario
24  for fila in range(filas):
25      fila_arreglo = []
26      for columna in range(columnas):
27          fila_arreglo.append(int(input("Ingresa el número de la posición [" + str(fila) + "][" + str(columna) + "]: ")))
28      matriz_2.append(fila_arreglo)
29
30  # Sumar matrices
31  matriz_suma = []
32  for fila in range(filas):
33      fila_arreglo = []
34      for columna in range(columnas):
35          fila_arreglo.append(int(matriz[fila][columna]+matriz_2[fila][columna]))
36      matriz_suma.append(fila_arreglo)
37  # Matriz
38  print("MATRIZ 1")
39  for i in matriz:
40      for j in i:

```

```
41     print(j, end=" ")
42     print()
43     # Matriz 2
44     print("MATRIZ 2")
45     for i in matriz_2:
46         for j in i:
47             print(j, end=" ")
48         print()
49
50     # Imprime la suma de las matrices
51     print("MATRICES SUMADAS")
52     for i in matriz_suma:
53         for j in i:
54             print(j, end=" ")
55         print()
```

### EJECUCIÓN:

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/Local/Microsoft/WindowsApps/python3.11.exe "matriz.py.py"
Introduzca el número de filas de las matrices: 2
Introduzca el número de columnas de la matrices: 3
Ingresar los elementos de la matriz 1
Ingrese el número de la posición [0][0]: 1
Ingrese el número de la posición [0][1]: 5
Ingrese el número de la posición [0][2]: 9
Ingrese el número de la posición [1][0]: 6
Ingrese el número de la posición [1][1]: 2
Ingrese el número de la posición [1][2]: 4
Ingresar los elementos de la matriz 2
Ingrese el número de la posición [0][0]: 8
Ingrese el número de la posición [0][1]: 1
Ingrese el número de la posición [0][2]: 3
Ingrese el número de la posición [1][0]: 6
Ingrese el número de la posición [1][1]: 8
Ingrese el número de la posición [1][2]: 7
MATRIZ 1
1 5 9
6 2 4
MATRIZ 2
8 1 3
6 8 7
MATRICES SUMADAS
9 6 12
12 10 11
PS E:\Documentos\Algoritmos\Deberes\Deber 10> |
```

Se solicita al usuario que inicialmente inserte tanto el número de filas como columnas, para ambas matrices ya que en esa relación para poder sumarlas deberían ser iguales, al insertar esos valores, el usuario insertara también los dígitos de cada matriz, mostrando en pantalla la matriz uno la dos y la suma de ambas.

11. Debido a la alta demanda y limitada oferta de cupos para el curso pre-politécnico de la EPN, se realiza el sorteo de cupos entre los aspirantes. Los estudiantes que han sido abanderados o escoltas en sus colegios, ingresan sin sorteo.



Se requiere un programa para realizar el registro de los aspirantes que participaran en el sorteo, ingresando su cédula, nombre, condecoración (1:abanderado/escolta, 0: ninguna).

De los  $n$  cupos disponibles, se les descuenta la cantidad de estudiantes que tienen condecoración, y se sortean los  $m$  cupos restantes entre los estudiantes.

Muestre la lista de todos estudiantes seleccionados.

Cédula	Nombre	Condecoración
0912345678	Juan Pedro Moreno	0
0987654321	Maria Ana Campos	1
...	...	...

NOTA:  $n$  se ingresa al inicio y  $m$  se cuenta luego del registro de aspirantes

“El sorteo de cupos para el ‘pre’ generó alegría y decepción”. [www.eluniverso.com](http://www.eluniverso.com). Enero 21 del 2010.

```

Deber > #Ejercicio_11_Cupos.py > sortear_cupos
1  # Ejercicios Tarea Josue Guerra, Carlos Perez, Richard Soria
2  # Cupos disponibles
3  import random
4
5  def registrar_aspirante():
6      cedula = input("Ingrese la cédula del aspirante: ")
7      nombre = input("Ingrese el nombre del aspirante: ")
8      condecoracion = int(input("¿Tiene condecoración? (1: Sí, 0: No): "))
9      print(" ")
10     return {'cedula': cedula, 'nombre': nombre, 'condecoracion': condecoracion}
11
12 def sortear_cupos(aspirantes, cupos):
13     seleccionados = []
14
15     condecorados = [aspirante for aspirante in aspirantes if aspirante['condecoracion'] == 1]
16
17     seleccionados.extend(condecorados)
18
19     cupos_restantes = cupos - len(condecorados)
20
21     restantes = [aspirante for aspirante in aspirantes if aspirante['condecoracion'] == 0]
22     seleccionados.extend(random.sample(restantes, cupos_restantes))
23
24     return seleccionados
25
26 def mostrar_seleccionados(seleccionados):
27     print("\nLista de estudiantes seleccionados:")
28     print("Cédula\tNombre\tCondecoración")
29     for estudiante in seleccionados:
30         print(f"{estudiante['cedula']}\t{estudiante['nombre']}\t{estudiante['condecoracion']}")

```

```

31
32 def main():
33     n_cupos = int(input("Ingrese la cantidad total de cupos disponibles: "))
34     m_cupos = int(input("Ingrese la cantidad de cupos a sortear: "))
35
36     aspirantes = []
37     # Registro de aspirantes
38     num_aspirantes = int(input("Ingrese la cantidad de aspirantes: "))
39     for _ in range(num_aspirantes):
40         aspirante = registrar_aspirante()
41         aspirantes.append(aspirante)
42
43     seleccionados = sortear_cupos(aspirantes, m_cupos)
44
45     # Mostrar la lista de seleccionados
46     mostrar_seleccionados(seleccionados)
47 if __name__ == "__main__":
48     main()

```

## EJECUCIÓN:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/Local/Python/Python39-64/Python.exe .py
Ingrese la cantidad total de cupos disponibles: 8
Ingrese la cantidad de cupos a sortear: 5
Ingrese la cantidad de aspirantes: 6
Ingrese la cédula del aspirante: 1752370344
Ingrese el nombre del aspirante: JOSUE
¿Tiene condecoración? (1: Sí, 0: No): 1

Ingrese la cédula del aspirante: 1752864928
Ingrese el nombre del aspirante: EDUARD
¿Tiene condecoración? (1: Sí, 0: No): 1

Ingrese la cédula del aspirante: 1752898546
Ingrese el nombre del aspirante: CARLOS
¿Tiene condecoración? (1: Sí, 0: No): 0

Ingrese la cédula del aspirante: 1754542863
Ingrese el nombre del aspirante: RICHARD
¿Tiene condecoración? (1: Sí, 0: No): 1

Ingrese la cédula del aspirante: 1658249348
Ingrese el nombre del aspirante: ADRIAN
¿Tiene condecoración? (1: Sí, 0: No): 0

Ingrese la cédula del aspirante: 1574824963
Ingrese el nombre del aspirante: MONSERATH
¿Tiene condecoración? (1: Sí, 0: No): 0

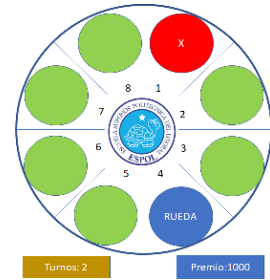
Lista de estudiantes seleccionados:
Cédula      Nombre      Condecoración
1752370344   JOSUE       1
1752864928   EDUARD      1
1754542863   RICHARD     1
1574824963   MONSERATH   0
1658249348   ADRIAN      0
PS E:\Documentos\Algoritmos\Deberes\Deber 10> 

```

## 12. Escriba un programa que implemente el “Juego de las Ruedas”.

Para esto genere aleatoriamente una lista de 15 elementos, donde cuatro elementos deben decir “Rueda” y los otros once “X”.

Luego el programa deberá pedirle al jugador que ingrese por teclado índices entre 0 y 14 (validar) correspondientes a la posición de la rueda.



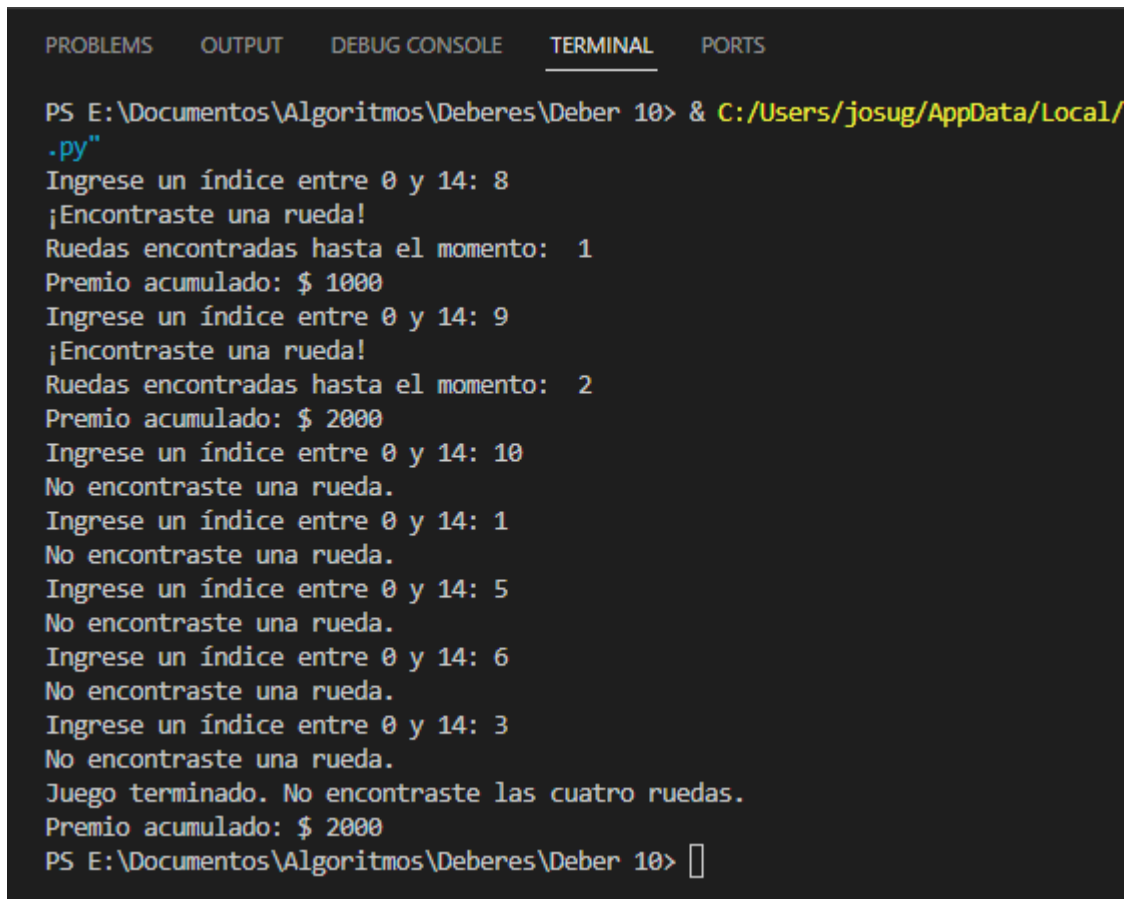
- Asuma que el jugador siempre ingresa índices distintos.
- Si el índice ingresado por el usuario corresponde al de una “Rueda”, gana \$1000.
- Si el jugador encuentra las cuatro “Ruedas” se gana un carro.
- El jugador tiene siete intentos para hallar las cuatro “Ruedas”.
- El juego termina cuando encuentra las cuatro “Ruedas” o ha usado todos los intentos.

```

Deber > #Ejercicio_12_rueda.py > jugar
1  # Ejercicios Tarea Josue Guerra, Carlos Perez, Richard Soria
2  # Juego de la rueda
3  import random
4
5  def generar_lista():
6      lista = ["X"] * 11 + ["Rueda"] * 4
7      random.shuffle(lista)
8      return lista
9
10 def jugar():
11     lista = generar_lista()
12     intentos = 7
13     ruedas_encontradas = 0
14     premio_total = 0
15     while intentos > 0 and ruedas_encontradas < 4:
16         indice = int(input("Ingrese un índice entre 0 y 14: "))
17         if indice < 0 or indice > 14:
18             print("Índice inválido. Intente nuevamente.")
19             continue
20         if lista[indice] == "Rueda":
21             ruedas_encontradas += 1
22             premio_total += 1000
23             print("¡Encontraste una rueda!")
24             print("Ruedas encontradas hasta el momento: ",ruedas_encontradas)
25             print("Premio acumulado: $",premio_total)
26         else:
27             print("No encontraste una rueda.")
28
29         intentos -= 1
30
31     if ruedas_encontradas == 4:
32         print(f"¡Ganaste un carro!")
33     else:
34         print("Juego terminado. No encontraste las cuatro ruedas.")
35         print("Premio acumulado: $",premio_total)
36
37     jugar()
  
```

En cada intento muestre por pantalla el número total de “Ruedas” encontradas hasta el momento. Al final muestre el premio que recibe el jugador (cantidad de dólares o la palabra “carro” si encontró las cuatro ruedas).

EJECUCIÓN:



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/Local/Python/Python39-64/Python.exe E:\Documentos\Algoritmos\Deberes\Deber 10.py
Ingrese un índice entre 0 y 14: 8
¡Encontraste una rueda!
Ruedas encontradas hasta el momento: 1
Premio acumulado: $ 1000
Ingrese un índice entre 0 y 14: 9
¡Encontraste una rueda!
Ruedas encontradas hasta el momento: 2
Premio acumulado: $ 2000
Ingrese un índice entre 0 y 14: 10
No encontraste una rueda.
Ingrese un índice entre 0 y 14: 1
No encontraste una rueda.
Ingrese un índice entre 0 y 14: 5
No encontraste una rueda.
Ingrese un índice entre 0 y 14: 6
No encontraste una rueda.
Ingrese un índice entre 0 y 14: 3
No encontraste una rueda.
Juego terminado. No encontraste las cuatro ruedas.
Premio acumulado: $ 2000
PS E:\Documentos\Algoritmos\Deberes\Deber 10> 
```

Se intentó varias veces para comprobar que el programa establezca correctamente cada función, además cuando el usuario gana también se va acumulando el premio que es en dinero si el participante acierta con el índice del número, tomando en cuenta que en cada ejecución los índices ganadores irán cambiando ya que el premio se almacena en diferentes números generados con la biblioteca random, y esto permite que no solo este en ciertos números predeterminados, más bien rotan entre el rango de 14.



## Parte III

### Consulta:

#### Algoritmos de búsqueda binaria

La búsqueda binaria es un algoritmo eficiente para encontrar un elemento en una lista ordenada de elementos. Funciona al dividir repetidamente a la mitad la porción de la lista que podría contener al elemento, hasta reducir las ubicaciones posibles a solo una.

Un ejemplo de ello es el siguiente: un usuario que tiene una lista ordenada de elementos, como una lista de números. Si desea encontrar un número específico en esta lista, la búsqueda binaria ofrece una estrategia inteligente y eficiente. En lugar de comenzar desde el principio de la lista, el algoritmo comienza en el medio. Si el número buscado es mayor que el número en el medio de la lista, el usuario sabe que no puede estar en la mitad inferior de la lista, por lo que descarta esa mitad y se queda solo con la mitad superior.

Luego, el usuario vuelve a calcular el punto medio de la mitad superior y repite el proceso. Si el número buscado es menor que el número en el medio de esta nueva mitad descarta la mitad superior y se queda solo con la mitad inferior. Este proceso de división por la mitad se repite hasta que se encuentra el número deseado o hasta que solo queda un elemento en la lista.

Este enfoque es eficiente porque en cada paso se elimina la mitad de los elementos posibles en los que podría estar el número buscado. Esto contrasta con la búsqueda lineal, donde podría ser necesario revisar cada elemento de la lista uno por uno. En resumen, la búsqueda binaria ofrece una manera inteligente y rápida de encontrar un elemento en una lista ordenada, ya que reduce rápidamente el espacio de búsqueda a la mitad en cada paso. [1]

#### RESUMEN

La búsqueda binaria es un algoritmo eficiente para encontrar un elemento en una lista ordenada. Funciona dividiendo repetidamente la lista en mitades y descartando la mitad donde no puede estar el elemento buscado, hasta reducir las ubicaciones posibles a solo una. Por ejemplo, si se busca un número en una lista ordenada, la búsqueda binaria comienza en el medio y ajusta la búsqueda según el valor del elemento en ese punto. Este proceso se repite hasta encontrar el elemento deseado o hasta que solo quede un elemento en la lista. Este enfoque es más rápido que la búsqueda lineal, ya que elimina la mitad de los elementos posibles en cada paso, lo que lo hace eficiente para listas grandes.

Para demostrar mejor se realizan dos ejemplos, los cuales se presentan en la siguiente página

## EJEMPLOS

## EJEMPLO 1:

```
Consulta > #Ejercicio1_BusBin.py > ...
1  # Ejercicios Josue Guerra, Carlos Perez, Richard Soria
2  # Búsqueda binaria
3  def busqueda_binaria(lista, numero):
4      inicio = 0
5      fin = len(lista) - 1
6
7      while inicio <= fin:
8          medio = (inicio + fin) // 2
9
10         if lista[medio] == numero:
11             return medio
12         elif lista[medio] < numero:
13             inicio = medio + 1
14         else:
15             fin = medio - 1
16     return -1
17
18 # Ejemplo de uso
19 lista = [1, 3, 5, 7, 9, 11, 13]
20 numero_buscar = int(input("Indique el valor a buscar: "))
21 resultado = busqueda_binaria(lista, numero_buscar)
22
23 if resultado != -1:
24     print("El número", numero_buscar, "se encuentra en el índice", resultado)
25 else:
26     print("El número", numero_buscar, "no se encuentra en la lista")

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/Local/Microsoft/WindowsApps/PythonSoftwareFoundation.Python.3.9.0.0_qbk9qtczc3c3nquar5wzq6v2p70/python.exe in.py
Indique el valor a buscar: 3
El número 3 se encuentra en el índice 1
PS E:\Documentos\Algoritmos\Deberes\Deber 10> 
```

Se declara una lista predefinida donde el usuario inserta el valor a buscar, tomando en cuenta que el programa lee la posición desde el índice 0, así que el primer número de la lista tendría esa posición y según eso los siguientes tomaran la posición de 1, 2, entre los demás

## EJEMPLO 2:

```

Consulta > #Ejercicio2_BusBin_2.py > ...
1  # Ejercicios Josue Guerra, Carlos Perez, Richard Soria
2  # Ejemplo2
3  # Búsqueda Binaria
4
5  def busqueda_binaria_recursiva(lista, valor, izq, der):
6      if izq > der:
7          return -1
8
9      medio = (izq + der) // 2
10
11     if lista[medio] == valor:
12         return medio
13     elif lista[medio] < valor:
14         return busqueda_binaria_recursiva(lista, valor, medio + 1, der)
15     else:
16         return busqueda_binaria_recursiva(lista, valor, izq, medio - 1)
17
18 # Ejemplo de uso
19 lista = [1, 3, 5, 7, 9, 11, 13]
20 while True:
21     valor = int(input("Inserte el valor a encontrar (0 para salir): ")) # Convertir la entrada a entero
22
23     if valor == 0:
24         break
25     resultado = busqueda_binaria_recursiva(lista, valor, 0, len(lista) - 1)
26     if resultado != -1:
27         print("El valor", valor, "se encuentra en el índice", resultado)
28     else:
29         print("El valor", valor, "no se encuentra en la lista")
30
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
PS E:\Documentos\Algoritmos\Deberes\Deber 10> & C:/Users/josug/AppData/Local/Microsoft/WindowsApps/python3.11.exe "e:/Document
in_2.py"
Inserte el valor a encontrar (0 para salir): 1
El valor 1 se encuentra en el índice 0
Inserte el valor a encontrar (0 para salir): 5
El valor 5 se encuentra en el índice 2
Inserte el valor a encontrar (0 para salir): 6
El valor 6 no se encuentra en la lista
Inserte el valor a encontrar (0 para salir): 0
PS E:\Documentos\Algoritmos\Deberes\Deber 10>

```

## ENTREGABLES:

- Una vez culminada tu tarea, capturar las pantallas de la ejecución del problema con tus datos y súbela en el apartado del aula virtual "S11-Tarea
- Subir los ejercicios al git o al drive y entrega la url de los archivos .py o, a su vez, entregue el archivo.
- Recordar que el nombre del archivo deberá ser: Tarea8\_Algoritmos\_2023B\_NApellido(de todos los integrantes)

## RECURSOS NECESARIOS

- Acceso a Internet.
- Imaginación.
- VSC

## ENLACE:

### Enlace GitHub:

<https://github.com/JosueGuerra2023B/Estructuras-Datos2023B/tree/master/Deberes/Deber%2010>

## RECOMENDACIONES

1. Organiza tu código en funciones claras y bien definidas.
2. Reutiliza código existente para evitar duplicación y mejorar la legibilidad.
3. Documenta adecuadamente tus funciones para facilitar la colaboración y el mantenimiento del código.
4. Simplifica las recomendaciones para un código más eficiente.

## CONCLUSIÓN

En resumen, el uso de funciones en Python es fundamental para mejorar el modularidad y la reutilización de código en nuestros programas. Al dividir el código en bloques más pequeños y manejables, podemos evitar la repetición innecesaria de líneas de código y facilitar el mantenimiento a largo plazo. Además, las funciones nos permiten encapsular lógica específica en un solo lugar, lo que facilita la depuración y el desarrollo de programas más eficientes.

## 1 Bibliografía

- [1 Khan Academy, «Khan Academy,» 2 Octubre 2015. [En línea]. Available:  
] <https://es.khanacademy.org/computing/computer-science/algorithms/binary-search/a/binary-search#:~:text=La%20b%C3%BAqueda%20binaria%20es%20un,ubicaciones%20posibles%20a%20solo%20una..> [Último acceso: 1 Febrero 2024].
- [2 M. ATEHORT, marzo 2002. [En línea]. Available: <http://goo.gl/HbbfD>.  
]