

# Tecnológico José Mario molina Pasquel y Henríquez UA Zapotlanejo

## Manual del programador

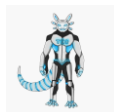


**Carrera:** Ingeniería en Informática

**Materia:** Estructura de datos

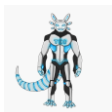
**Docente:** Osvaldo Rene Rojo Roa

**Alumno:** Oscar Alejandro Alvarez Aceves-240111676



## INDICE

PARCIAL 1 .....	3
Programa 1: Pedir un número, calcular la raíz y elevar al cuadrado .....	3
Programa 2: Arreglos y listas .....	4
Programa 3: Ingresar datos a arreglo .....	5
Programa 4: datos en arreglos y listas.....	6
Tarea 1: Ingresar datos a un arreglo o una lista .....	7
Programa 6: valores en listas con validaciones .....	9
Programa 7: información de 5 personas .....	10
Programa 8: Mayúsculas, minúsculas y espacios en una cadena .....	12
Tarea 2: lista de cadenas con condiciones.....	13
Programa 9: Resolución tarea 2 .....	14
Repaso 1 .....	15
Repaso 2 .....	16
Repaso 3 .....	17



## PARCIAL 1

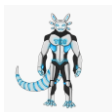
Programa 1: Pedir un número, calcular la raíz y elevar al cuadrado

```
1 a = int(input('Escribe un numero: ')) # Ask the user to enter a number | Solicita al usuario que ingrese un número
2 print(a**2) # Calculate the power of the number (a²) | Calcula la potencia del número (a²)
3 print(a**(1/2)) # Calculate the square root of the number | Calcula la raíz cuadrada del número
4
5 # Operadores lógicos / Logical operators
6 '''
7 +, -, * # Addition, subtraction, multiplication | Suma, resta, multiplicación
8 / (exacta con decimales), //(sin decimales), ** (Potencia o elevar a), mod, and, or
9 / division with decimals, // integer division, ** exponentiation, mod modulus, and/or logical operators
10 / división exacta con decimales, // división entera, ** potencia, mod módulo, and/or operadores lógicos
11 '''
12
13 # Operadores diferenciales / Relational operators
14 '''
15 <, >, <=, >=, !=, ==, not
16 < less than, > greater than, <> not equal, >= greater or equal, <= less or equal, != not equal, == equal, not negation
17 < menor que, > mayor que, <> diferente, >= mayor o igual, <= menor o igual, != distinto, == igual, not negación
18 '''
19
```

Ilustración 1 Código Programa 1

```
PS C:\Users\oscar\Dropbox\UNIVERSIDAD\Te
ticasParcial1> & C:\Users\oscar\AppData\
scar\Dropbox\UNIVERSIDAD/Tercer semestre
/Prog01.py"
Escribe un numero: 10
100
3.1622776601683795
PS C:\Users\oscar\Dropbox\UNIVERSIDAD\Te
ticasParcial1> █
```

Ilustración 2 ejecución programa 1



## Programa 2: Arreglos y listas

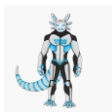
Este programa muestra las diferencias entre un arreglo y una lista, también enseña la manera correcta de inicializarlas

```
1 a = [10] # Arreglo | Array
2 b = [ ] # Lista | List
3
4 a[0] = 10 # Asigna el valor 10 a la primera posición del arreglo | Assigns value 10 to the first position of the array
5
6 b = {'Hola', 10, 100.05, False, 'm', {1, 2, 3, 4, 5, 6, 7}}
7 # Una lista permite guardar cualquier tipo de dato sin importar su categoría
8 # A list allows you to store any type of data regardless of its category
9
10 # b.append(10) # Agrega un elemento al final de la lista | Adds an element to the end of the list
11
12 ## Ciclos y condiciones | Loops and conditions
13 if(len(a) > len(b)): # Compara el tamaño de las listas | Compares the size of the lists
14     print('A es mayor') # Imprime que A es mayor | Prints that A is greater
15 else:
16     print('B es mayor') # Imprime que B es mayor | Prints that B is greater
17
18 # Ciclo for para recorrer la lista | For loop to iterate through the list
19 for i in a:
20     print(a) # Imprime la lista completa en cada iteración | Prints the whole list on each iteration
21
```

Ilustración 3 Código programa 2

```
C:\Users\oscar\AppData\Local\Microsoft\Windows
mestre/Estructura de datos/Primer semestre/Pr
B es mayor
[10]
PS C:\Users\oscar\Dropbox\UNIVERSIDAD\Tercer
```

Ilustración 4 Ejecución programa 2



### Programa 3: Ingresar datos a arreglo

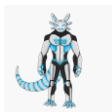
Este programa enseña como añadir valores en un arreglo respetando la posición de entrada

```
1 # Hacer un programa que lea 10 números y los almacene en un arreglo
2 # Make a program that reads 10 numbers and stores them in an array
3
4 a = [0,0,0,0,0,0,0,0,0,0] # Se inicializa un arreglo con 10 elementos en 0 | Initialize an array with 10 elements set to 0
5
6 # Ciclo para leer 10 números y guardarlos en el arreglo | Loop to read 10 numbers and store them in the array
7 for i in range(0,10):
8     a[i] = int(input(f'Ingresa un numero \n')) # Solicita un número y lo guarda en la posición correspondiente | Asks for a number and stores it in the corresponding position
9
10 # Ciclo para imprimir los números almacenados en el arreglo | Loop to print the stored numbers in the array
11 for i in a:
12     print(i) # Muestra cada número en una nueva línea | Displays each number on a new line
13
```

Ilustración 5 Código programa 3

```
10
10
10
10
10
10
10
10
10
PS C:\Users\oscar\Dropbox\
```

Ilustración 6 ejecución programa 3



## Programa 4: datos en arreglos y listas

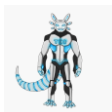
Este programa enseña una forma básica de validar que los datos que se ingresaran a una lista son del tipo entero, al final mostrara el total de la suma de los datos aceptados

```
1 # Hacer un programa que lea 10 números y los almacene en una lista
2 # Create a program that reads 10 numbers and stores them in a list
3
4 a = [] # Lista donde se guardarán los números | List where the numbers will be stored
5 s = 0 # Variable para almacenar la suma | Variable to store the sum
6 n = 0 # Contador para controlar la cantidad de números | Counter to control how many numbers are entered
7 numeros = "0,1,2,3,4,5,6,7,8,9"; # Define el rango de datos que se pueden usar | Defines the range of allowed digits
8
9 # Ciclo while que continúa hasta que se ingresen 10 números válidos | While loop that continues until 10 valid numbers are entered
10 while(n < 10):
11     b = input('Escribe un numero: ') # Solicita un número al usuario | Asks the user for a number
12     x = 0 # Contador de dígitos válidos | Counter for valid digits
13
14     for i in b: # Recorre cada carácter ingresado | Iterates through each entered character
15         # if (ord(i) >= 48 and ord(i) <= 57): # El ord se utiliza para obtener el valor ASCII | The ord function is used to get the ASCII value
16             if i in numeros: # Verifica si el carácter es un número válido | Checks if the character is a valid number
17                 x += 1 # Incrementa el contador si es válido | Increments the counter if valid
18
19     if len(b) == x: # Compara si todos los caracteres son números | Checks if all characters are valid digits
20         a.append(int(b)) # Convierte a entero el valor y lo agrega a la lista | Converts the value to integer and appends it to the list
21         n += 1 # Incrementa el contador | Increments the counter
22     else:
23         print('El valor no es numero') # Muestra un mensaje si hay caracteres inválidos | Displays a message if invalid characters are detected
24
25 # Imprime todos los números de la lista | Prints all the numbers stored in the list
26 for i in a:
27     print(i) # Muestra cada número | Displays each number
28     s += i # Suma cada número a la variable acumuladora | Adds each number to the accumulator
29
30 # Muestra la suma total de los números ingresados | Displays the total sum of the entered numbers
31 print(f'La suma es: {s}')
```

Ilustración 7 Código programa 4

```
10
10
10
10
10
10
10
10
La suma es: 100
PS C:\Users\oscar\Dropbox\UNIVE
```

Ilustración 8 ejecución programa 4



## Tarea 1: Ingresar datos a un arreglo o una lista

Este programa enseña una manera básica de validar si un dato es numérico o de caracteres, si es numérico lo almacenara en un arreglo, por otro lado, si es carácter lo guardara en una lista, el programa hará esto hasta que las longitudes del arreglo y lista sumen un total de 10

```

1 #Hacer un programa que lea 10 datos, si el dato es un numero se almacenara en un arreglo
2 # si es un caracter o caracteres se metera en una lista, cuando finalice el programa
3 # nos mostrara cuantos elementos numericos y cuantos caracteres hay en cada estructuras
4
5 car = 0 # EN: Variable declared but unused / ES: Variable declarada pero no utilizada
6 Arreglo = 0 # EN: Counter for total entered elements / ES: Contador para los elementos totales ingresados
7 n = 0 # EN: Counter for total entered elements / ES: Contador para los elementos ingresados
8 a = [0,0,0,0,0,0,0,0,0,0] # EN: Initialize list 'a' with one numeric element (10) / ES: Inicializa la lista 'a' con un elemento numérico (10)
9 l = [] # EN: Initialize empty list 'l' to store alphabetic characters / ES: Inicializa la lista vacía 'l' para guardar caracteres alfabéticos
10
11 while (n < 10): # EN: Loop until 10 valid elements are entered / ES: Bucle hasta que se ingresen 10 elementos válidos
12     b = input('Ingresa un dato: ') # EN: Request input from user / ES: Solicita un dato al usuario
13
14     if b.isdigit(): # EN: Check if input is numeric / ES: Verifica si el dato ingresado es numérico
15         a[Arreglo]=(b) # EN: Add numeric input to list 'a' / ES: Agrega el dato numérico a la lista 'a'
16         Arreglo +=1 # EN: Increase numeric counter / ES: Incrementa el contador de numéricos
17         n += 1 # EN: Increment counter / ES: Incrementa el contador
18     elif b.isalpha(): # EN: Check if input is alphabetic / ES: Verifica si el dato ingresado es alfabético
19         l.append(b) # EN: Add alphabetic input to list 'l' / ES: Agrega el dato alfabético a la lista 'l'
20         n += 1 # EN: Increment counter / ES: Incrementa el contador
21     else:
22         print('No valido') # EN: Show error message for invalid input / ES: Muestra mensaje de error para datos no válidos
23
24 # ---- Final results ----
25 print(f'El total de elementos numericos guardados es {(Arreglo)}')
26 # EN: Show total numeric elements stored / ES: Muestra el total de elementos numéricos guardados
27 print(f'El total de caracteres guardados es {len(l)}')
28 # EN: Show total alphabetic elements stored / ES: Muestra el total de caracteres guardados
29 """print(f'Elementos numericos guardados (a)')
30 # EN: Display list of numeric elements / ES: Muestra la lista de elementos numéricos
31 print(f'Elementos de caracteres guardados (l)')
32 # EN: Display list of alphabetic elements / ES: Muestra la lista de elementos alfabéticos"""
33

```

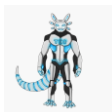
Ilustración 9 Código tarea 1

```

Ingresa un dato: hola
Ingresa un dato: que
Ingresa un dato: tal
Ingresa un dato: 3
Ingresa un dato: 2
El total de elementos numericos guardados es 6
El total de caracteres guardados es 4
PS C:\Users\oscar\Dropbox\UNIVERSIDAD\Tercer semestre\TareasParciales>

```

Ilustración 10 ejecución tarea 1

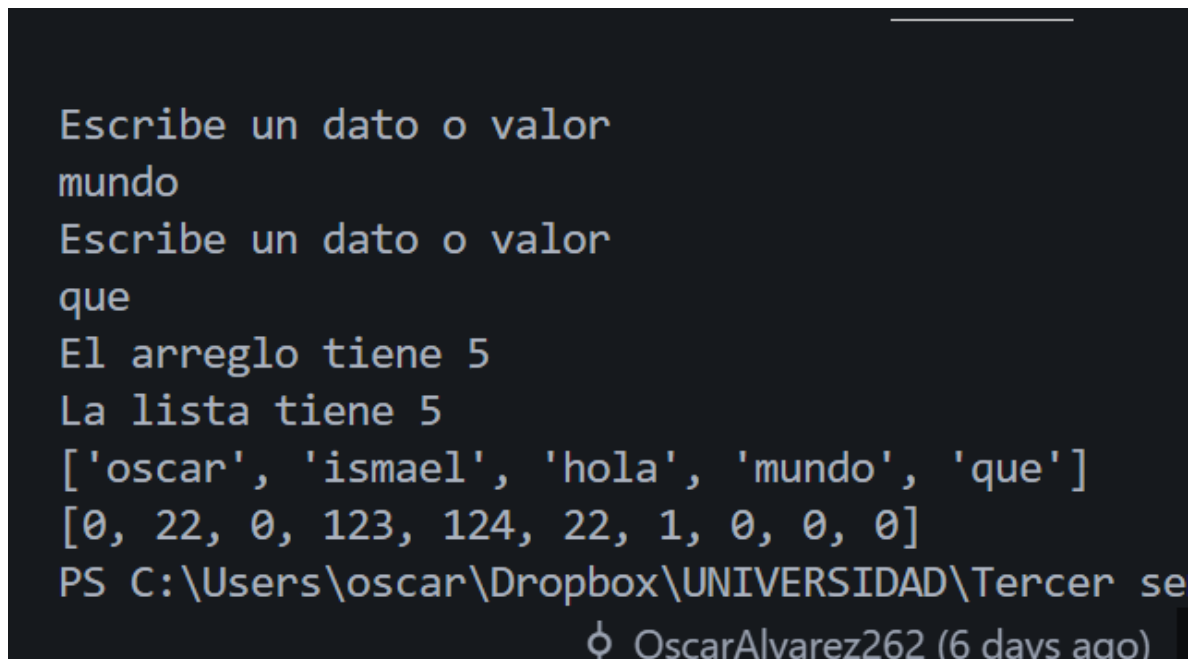


## Programa 5: resolución tarea 1

Este programa enseña una manera alternativo de resolver la problemática de la tarea 1

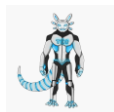
```
1 arr = [0,0,0,0,0,0,0,0,0,0] # Arreglo para almacenar números | Array to store numbers
2 car = [] # Lista para almacenar cadenas | List to store strings
3 c = 0 # Contador para la posición del arreglo | Counter for array position
4 c2 = 0 # Contador de números válidos en el arreglo | Counter for valid numbers in the array
5
6 # Bucle para leer hasta 10 datos | Loop to read up to 10 inputs
7 while (True):
8     a = input("Escribe un dato o valor \n") # Solicita un dato al usuario | Asks the user for a value
9
10    if a.isdigit(): # Verifica si es un número | Checks if the input is a number
11        arr[c] = int(a) # Convierte a entero y lo guarda en el arreglo | Converts to integer and stores in the array
12    elif a.isalpha(): # Verifica si es solo letras | Checks if the input contains only letters
13        car.append(a) # Agrega la cadena a la lista | Appends the string to the list
14
15    c += 1 # Incrementa el contador | Increments the counter
16    if c >= 10: # Sale del bucle después de 10 entradas | Exits the loop after 10 inputs
17        break
18
19 # Cuenta cuántos números válidos hay en el arreglo | Counts how many valid numbers are in the array
20 for i in arr:
21     if i != 0:
22         c2 += 1
23
24 print(f'El arreglo tiene {c2}') # Muestra la cantidad de números | Displays the number of numbers
25 print(f'La lista tiene {len(car)} ') # Muestra la cantidad de cadenas | Displays the number of strings
26
27 print(car) # Imprime la lista de cadenas | Prints the list of strings
28 print(arr) # Imprime el arreglo de números | Prints the array of numbers
29
30
```

Ilustración 11 Código programa 5



```
Escribe un dato o valor
mundo
Escribe un dato o valor
que
El arreglo tiene 5
La lista tiene 5
['oscar', 'ismael', 'hola', 'mundo', 'que']
[0, 22, 0, 123, 124, 22, 1, 0, 0, 0]
PS C:\Users\oscar\Dropbox\UNIVERSIDAD\Tercer se
OscarAlvarez262 (6 days ago)
```

Ilustración 12 ejecución programa 5



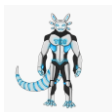


## Programa 6: valores en listas con validaciones

Este programa enseña el uso de funciones para realizar una validación si un dato ingresado es tipo numérico o carácter almacenando en un arreglo o lista respectivamente, mostrando al final el total y cuales datos fueron almacenados

```
1 def hola(): # Definición de método o función | Method or function definition
2     c = 0 # Contador para controlar la posición en el arreglo | Counter to control the position in the array
3     while (True): # Bucle infinito hasta que se ingrese 10 datos | Infinite loop until 10 inputs are given
4         a = input('Escribe un dato o valor \n') # Solicita un dato al usuario | Asks the user for a value
5
6         if a.isdigit(): # Verifica si la entrada es un número | Checks if the input is a number
7             arr[c] = int(a) # Convierte a entero y lo guarda en el arreglo | Converts to integer and stores it in the array
8         elif a.isalpha(): # Verifica si la entrada es solo letras | Checks if the input is alphabetic
9             car.append(a) # Agrega la cadena a la lista car | Appends the string to the car list
10
11         c += 1 # Incrementa el contador | Increments the counter
12
13         if c >= 10: # Condición para detener el ciclo después de 10 entradas | Stops the loop after 10 inputs
14             break
15
16     resultados() # Llama a la función para mostrar los resultados | Calls the function to display results
17
18
19 def resultados(): # Definición de método o función | Method or function definition
20     c2 = 0 # Contador de valores numéricos válidos en el arreglo | Counter for valid numeric values in the array
21     print(f'La lista tiene {len(car)} ') # Muestra la cantidad de cadenas guardadas | Displays the number of stored strings
22
23     for i in arr: # Recorre el arreglo numérico | Iterates over the numeric array
24         if i != 0: # Verifica si el valor no es cero | Checks if the value is not zero
25             c2 += 1 # Incrementa el contador | Increments the counter
26
27     print(f'El arreglo tiene {c2}') # Muestra la cantidad de números guardados | Displays the number of stored numbers
28     print(car) # Imprime la lista de cadenas | Prints the list of strings
29     print(arr) # Imprime el arreglo de números | Prints the array of numbers
30
31
32 arr = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0] # Arreglo inicializado con 10 ceros | Array initialized with 10 zeros
33 car = [] # Lista para almacenar cadenas | List to store strings
34
35 if __name__ == "__main__": # Método principal | Main method
36     hola() # Llama a la función principal | Calls the main function
37
```

Ilustración 13 código programa 6



```
tal
Escribe un dato o valor
133
La lista tiene 6
El arreglo tiene 4
['hola', 'hola', 'hola', 'hola', 'que', 'tal']
[0, 0, 3, 0, 10, 0, 45, 0, 0, 133]
PS C:\Users\oscar\Dropbox\UNIVERSIDAD\Tercer sem
ticasParcial1>
```

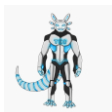
Ilustración 14 ejecución programa 6

### Programa 7: información de 5 personas

Este programa enseña el cómo almacenar múltiples variables en una posición de una lista

```
1 # Hacer un programa que lea nombre, edad y sexo de 5 personas
2 # Make a program that reads name, age, and gender of 5 people
3
4 def lista(): # Definición de función | Function definition
5     c = 0 # Contador para controlar cuántas personas se registran | Counter to control how many people are registered
6     while True: # Bucle infinito hasta que se ingresen 5 personas | Infinite loop until 5 people are entered
7         n = input('Ingresa un nombre \n') # Solicita el nombre | Asks for the name
8         e = input('Ingresa su edad \n') # Solicita la edad | Asks for the age
9         g = input('Ingresa su sexo H o M \n').upper() # Solicita el sexo y lo convierte a mayúsculas | Asks for gender and converts it to uppercase
10
11         # Se guarda la información en la lista lisi como una cadena | Stores the information in lisi as a string
12         lisi.append(n + ', ' + e + ', ' + g)
13
14         c += 1 # Incrementa el contador | Increments the counter
15
16         if c >= 5: # Si ya se ingresaron 5 personas, sale del bucle | Stops the loop after 5 people are entered
17             break
18
19     resultado() # Llama a la función para mostrar resultados | Calls the function to display results
20
21
22 def resultado(): # Definición de función para mostrar resultados | Function definition to display results
23     print(lisi) # Imprime la lista con los datos registrados | Prints the list with the stored data
24
25
26 lisi = [] # Lista para almacenar nombre, edad y sexo | List to store name, age, and gender
27 lis2 = [] # Lista no utilizada actualmente | List not used currently
28
29 # Punto de inicio del programa | Program entry point
30 if __name__ == "__main__":
31     lista() # Llama a la función principal | Calls the main function
32
```

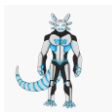
Ilustración 15 código programa 7



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Ingresa un nombre
sandra
Ingresa su edad
20
Ingresa su sexo H o M
M
['jose, 33, H', 'Maria, 12, M', 'oscar, 23, H', 'Diego, 19, H', 'sandra, 20, M']
PS C:\Users\oscar\Dropbox\UNIVERSIDAD\Tercer semestre\Estructura de datos\Primer semestre\Practicas\Practica 14>
```

Ilustración 16 ejecución programa 7



## Programa 8: Mayúsculas, minúsculas y espacios en una cadena

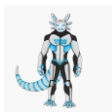
Este programa enseña la manera de hacer validaciones de características especiales incluyendo especificaciones y reglas en los datos que podrán ser almacenados en una lista

```
1 # Hacer un programa que lea una cadena y que muestre en pantalla
2 # cuántos números tiene, cuántas mayúsculas, cuántas minúsculas
3 # y cuántos espacios
4 # Make a program that reads a string and displays how many numbers, uppercase letters,
5 # lowercase letters, and spaces it contains
6
7 def inicio(): # Definición de función | Function definition
8     numeros = "0123456789" # Cadena de números válidos | String containing valid numbers
9     cn = 0 # Contador de números | Counter for numbers
10    e = 0 # Contador de espacios | Counter for spaces
11    min = 0 # Contador de minúsculas | Counter for lowercase letters
12    may = 0 # Contador de mayúsculas | Counter for uppercase letters
13
14    cadena = input('Escribe una cadena \n') # Solicita una cadena al usuario | Asks the user to enter a string
15
16    # Recorre cada carácter en la cadena | Iterates over each character in the string
17    for i in cadena:
18        if i in numeros: # Verifica si el carácter es un número | Checks if the character is a number
19            cn += 1 # Incrementa el contador de números | Increments the number counter
20
21        if i == ' ': # Verifica si el carácter es un espacio | Checks if the character is a space
22            e += 1 # Incrementa el contador de espacios | Increments the space counter
23
24        if ord(i) >= 97 and ord(i) <= 122: # Verifica si es una letra minúscula | Checks if it's a lowercase letter
25            min += 1 # Incrementa el contador de minúsculas | Increments lowercase counter
26
27        if ord(i) >= 65 and ord(i) <= 90: # Verifica si es una letra mayúscula | Checks if it's an uppercase letter
28            may += 1 # Incrementa el contador de mayúsculas | Increments uppercase counter
29
30    # Muestra los resultados finales | Displays the final results
31    print(f'Los numeros son {cn},\n los espacios: {e},\n las minusculas son: {min},\n las mayusculas son: {may}')
32
33
34 # Punto de inicio del programa | Program entry point
35 if __name__ == '__main__':
36     inicio() # Llama a la función principal | Calls the main function
37
38
```

Ilustración 17 Código programa 8

```
/Prog08.py"
Escribe una cadena
Hola que tal
Los numeros son 0,
los espacios: 2,
las minusculas son: 9,
las mayusculas son: 1
PS C:\Users\oscar\Dropbox\UNIVERS
ticasParcial1>
```

Ilustración 18 ejecución programa 8



## Tarea 2: lista de cadenas con condiciones

Este programa enseña la manera de hacer validaciones de características especiales incluyendo especificaciones y reglas en los datos que podrán ser almacenados en una lista

```

1 # hacer un programa que en una lista se introduzca cadenas de
2 # caracteres con las siguientes restricciones:
3 # 1.- las cadenas no tienen tener espacios
4 # 2.- la cadena solo puede tener mayuscula la primer letra
5 # 3.- obligatoriamente deben de tener todas las vocales.
6 # el programa no termina hasta que la lista tenga 5 elementos
7 def inicio():
8     voc = "AEIOU" # String containing vowels / Cadena con vocales (no usada)
9     lis = [] # list to store valid words / Lista para almacenar palabras válidas
10    ele = len(lis) # Variable initialized but unused / Variable inicializada pero no utilizada
11
12    # Loop until the list contains 5 valid words / Bucle hasta que la lista tenga 5 palabras válidas
13    while (len(lis) <= 4):
14        cad = input('Ingresa una cadena:\n') # Ask the user for input / Solicita al usuario ingresar una cadena
15        c = 0 # Counter for spaces / Contador de espacios
16        m = 0 # Counter for uppercase letters / Contador de letras mayúsculas
17        ca = 0 # Counter for vowel 'a' / Contador para la vocal 'a'
18        ce = 0 # Counter for vowel 'e' / Contador para la vocal 'e'
19        ci = 0 # Counter for vowel 'i' / Contador para la vocal 'i'
20        co = 0 # Counter for vowel 'o' / Contador para la vocal 'o'
21        cu = 0 # Counter for vowel 'u' / Contador para la vocal 'u'
22
23        # Iterate through each character in the string / Recorre cada carácter en la cadena
24        for i in cad:
25            if i == ' ':
26                c += 1 # Count spaces / Contar espacios
27
28            if ord(i) >= 65 and ord(i) <= 90: # Check if character is uppercase / Verificar si el carácter es mayúscula
29                m += 1
30
31            if ord(i) >= 65 and ord(i) <= 97: # Check if character is between 'A' and 'a' (possible mistake) / Verifica si el carácter está entre 'A' y 'a' (posible error)
32                ca += 1
33
34            if ord(i) == 69 or ord(i) == 101: # Count vowel 'e' / Contar vocal 'e'
35                ce += 1
36
37            if ord(i) == 73 or ord(i) == 105: # Count vowel 'i' / Contar vocal 'i'
38                ci += 1
39
40            if ord(i) == 79 or ord(i) == 111: # Count vowel 'o' / Contar vocal 'o'
41                co += 1
42
43            if ord(i) == 85 or ord(i) == 117: # Count vowel 'u' / Contar vocal 'u'
44                cu += 1
45
46        # ----- VALIDATIONS / VALIDACIONES -----
47        # Check if first letter is uppercase and only one uppercase is allowed / Verifica que la primera letra sea mayúscula y solo exista una mayúscula
48        if (m == 0) or (m == 1) and (ord(cad[0]) >= 65 and ord(cad[0]) <= 90):
49            if c >= 1:
50                print('No puede llevar espacios') # Invalid if spaces exist / Inválido si existen espacios
51
52            # Check if the word contains all vowels / Verifica si la palabra contiene todas las vocales
53            elif ca >= 1 and ce >= 1 and ci >= 1 and co >= 1 and cu >= 1:
54                lis.append(cad) # Save the valid word / Guarda la palabra válida
55                print("Palabra aceptada") # Accepted word / Palabra aceptada
56            else:
57                print('Error, la palabra debe de llevar todas las vocales')
58                # Error if missing any vowel / Error si falta alguna vocal
59
60        else:
61            print('Error, solo la primer letra puede ser mayuscula')
62            # Error if the first letter is lowercase or has multiple uppercase letters / Error si la primera letra es minúscula o si hay varias mayúsculas
63
64        # Show all saved valid words / Muestra todas las palabras válidas guardadas
65        print(f'Las palabras guardadas fueron {lis}')
66
67    # Program entry point / Punto de entrada del programa
68    if __name__ == '__main__':
69        inicio()
70

```

Ilustración 19 Código tarea 2

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

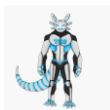
PORTS

```

Eaieou
"Palabra aceptada"
Ingresa una cadena:
mamemimomu
"Palabra aceptada"
Las palabras guardadas fueron ['Murcielago', 'Aeiou', 'aaaaaeiiiiooooouu', 'Eaieou', 'mamemimo
mu']
PS C:\Users\oscar\Dropbox\UNIVERSIDAD\Tercer semestre\Estructura de datos\Primer semestre\Prac
ticasParcial1>

```

Ilustración 20 ejecución tarea 2



## Programa 9: Resolución tarea 2

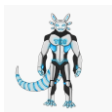
Este programa enseña la manera de hacer validaciones de características especiales incluyendo especificaciones y reglas en los datos que podrán ser almacenados en una lista

```

1 def minusculas(c1): # Verifica si todos los caracteres excepto el primero son minúsculas | Checks if all characters except the first are lowercase
2     cm = 0 # Contador de minúsculas | Counter for lowercase letters
3     for i in c1[1:]: # Recorre la cadena desde el segundo carácter | Iterates over the string from the second character
4         if ord(i) >= 97 and ord(i) <= 122: # Comprueba si el carácter es minúscula | Checks if the character is lowercase
5             cm += 1
6     if cm == len(c1) - 1: # Si todas excepto la primera son minúsculas | If all except the first are lowercase
7         print('La cadenas son minúsculas excepto la primera') # Mensaje de confirmación | Confirmation message
8     vocales(c1) # Llama a la función vocales para verificar que contenga todas las vocales | Calls vocales function to check all vowels
9 else:
10     print('La cadena no cumple') # Mensaje si no cumple la condición | Message if condition is not met
11
12 def vocales(cad): # Verifica si la cadena contiene todas las vocales | Checks if the string contains all vowels
13     ba = False # Bandera para la vocal 'a' | Flag for vowel 'a'
14     be = False # Bandera para la vocal 'e' | Flag for vowel 'e'
15     bi = False # Bandera para la vocal 'i' | Flag for vowel 'i'
16     bo = False # Bandera para la vocal 'o' | Flag for vowel 'o'
17     bu = False # Bandera para la vocal 'u' | Flag for vowel 'u'
18
19     if 'a' in cad or 'A' in cad:
20         ba = True
21     if 'e' in cad or 'E' in cad:
22         be = True
23     if 'i' in cad or 'I' in cad:
24         bi = True
25     if 'o' in cad or 'O' in cad:
26         bo = True
27     if 'u' in cad or 'U' in cad:
28         bu = True
29
30     # Si contiene todas las vocales, se agrega a la lista | If it contains all vowels, append to list
31     if ba and be and bi and bo and bu:
32         list.append(cad)
33
34 def leer(): # Función para leer y procesar la cadena | Function to read and process the string
35     ce = 0 # Contador de caracteres sin espacio | Counter for characters without spaces
36     nc = "" # Nueva cadena filtrada sin números | New filtered string without numbers
37     c = input('Introduce una cadena\n') # Solicita al usuario la cadena | Asks the user for a string
38
39     # Contar caracteres que no sean espacios | Count characters that are not spaces
40     for i in c:
41         if ord(i) != 32:
42             ce += 1
43
44     if ce == len(c): # Si no hay espacios | If there are no spaces
45         if c.isalpha(): # Si la cadena es solo letras | If string contains only letters
46             minusculas(c) # Llama a la función minusculas | Calls minusculas function
47         else: # Si contiene números u otros caracteres | If it contains numbers or other characters
48             for i in c:
49                 if ord(i) >= 48 and ord(i) <= 57: # Ignora números | Ignore numbers
50                     pass
51                 else:
52                     nc += i # Construye cadena filtrada | Builds filtered string
53             print(nc) # Muestra la cadena filtrada | Displays filtered string
54             minusculas(nc) # Llama a minusculas con la cadena filtrada | Calls minusculas with filtered string
55     else:
56         print('Error, hay espacios en la cadena') # Mensaje de error si hay espacios | Error message if spaces are present
57
58 list = [] # Lista para almacenar cadenas válidas | List to store valid strings
59
60 if __name__ == '__main__': # Punto de entrada del programa | Program entry point
61     while True:
62         leer() # Llama a la función leer repetidamente | Calls leer function repeatedly
63         if len(list) >= 5: # Sale del bucle cuando la lista tiene 5 elementos válidos | Exit loop when list has 5 valid strings
64             break
65

```

Ilustración 21 Código programa 9



## Repaso 1

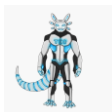
Este programa recopila los tipos de datos, el uso de rangos y las conversiones de datos ingresados

```
1  """Instrucciones de entrada y salida"""
2  #Print() o print(f)
3  # print('Hola mundo')
4  # print(f'Hola mundo {10}')
5
6  """Entrada de datos"""
7  #input
8  # input('Escribe un numero')#Se introducen SOLO letras
9  # #Casting para convertir a valores especificos
10 # f=0.0
11 # f = float(input('Ingresa un numero con decimales'))#Solo numeros flotantes
12 # e=0
13 # e= int(input('Escribe un numero'))#SOLO numeros enteros
14 # c =120
15 # print(str(c))#Convertir a cadena
16 # v=""
17 # v=str(c)
18 """NOTA: SOLO las variables que no se introducen por teclado se OBLIGAN a inicializarlas"""
19
20
21 # # Hacer un programa que sea nombre y precio de un producto, el programa
22 # # calculara el costo y precio de venta, el costo involucra el 12%
23 # y el iva 16%
24
25
26
27 for i in range(0, 5): # Repite el ciclo 5 veces | Loops 5 times
28     p = "" # Variable para almacenar el nombre del producto | Variable to store the product name
29     c = 0.0 # Precio bruto | Base price
30     u = 0.0 # Precio con IVA del 12% | Price with 12% tax
31     v = 0.0 # Precio de venta final con IVA + 16% | Final selling price with additional 16%
32
33     # Solicita el nombre del producto | Asks for the product name
34     p = input('Ingresa el nombre de un producto\n')
35
36     # Solicita el precio bruto del producto | Asks for the base price of the product
37     c = float(input(f'Ingresa el precio bruto del producto: {p}\n'))
38
39     # Calcula el precio con el 12% agregado | Calculates the price with an additional 12%
40     u = float(c * 1.12)
41
42     # Calcula el precio final de venta con otro 16% agregado | Calculates the final price with an extra 16%
43     v = (u * 1.16)
44
45     # Muestra el precio con el 12% de incremento | Displays the price after 12% increase
46     print(f'tu costo del producto: {p}, es de ${u:.2f}') # :.2f muestra 2 decimales | :.2f shows 2 decimals
47
48     # Muestra el precio final de venta | Displays the final selling price
49     print(f'tu precio de venta del producto: {p}, es de ${v:.2f}') # :.2f muestra 2 decimales | :.2f shows 2 decimals
```

Ilustración 22 Código repaso 1

```
mestre/Estructura de datos/Primer semestre/PracticasParcial1
Ingresa el nombre de un producto
papa
Ingresa el precio bruto del producto: papa
12
tu costo del producto: papa, es de $13.44
tu precio de venta del producto: papa, es de $15.59
Ingresa el nombre de un producto
```

Ilustración 23 ejecución repaso 1





## Repaso 2

Este programa enseña la manera de manipular datos tipos números ingresados para poder hacer potencias o raíces para poder realizar operaciones

```

1 # Hacer un programa que realice la operación de la fórmula general
2 # Make a program that performs the quadratic formula operation
3
4 a = 0 # Coeficiente A de la ecuación cuadrática | Coefficient A of the quadratic equation
5 b = 0 # Coeficiente B de la ecuación cuadrática | Coefficient B of the quadratic equation
6 c = 0 # Coeficiente C de la ecuación cuadrática | Coefficient C of the quadratic equation
7
8 p = 0 # Variable temporal para b² | Temporary variable for b squared
9 m = 0 # Variable temporal para 4ac | Temporary variable for 4ac
10 r = 0 # Discriminante b² - 4ac | Discriminant b² - 4ac
11
12 ra = 0.0 # Raíz cuadrada del discriminante | Square root of the discriminant
13 d = 0.0 # Denominador 2a | Denominator 2a
14 x1 = 0.0 # Primera solución de la ecuación cuadrática | First solution of the quadratic equation
15 x2 = 0.0 # Segunda solución de la ecuación cuadrática | Second solution of the quadratic equation
16
17
18 # Solicita los valores de A, B y C | Asks for the values of A, B, and C
19 a = int(input('Ingresa el valor de A: '))
20 b = int(input('Ingresa el valor de B: '))
21 c = int(input('Ingresa el valor de C: '))
22
23 # Calcula b² | Calculates b squared
24 p = b ** 2
25
26 # Calcula 4ac | Calculates 4ac
27 m = 4 * a * c
28
29 # Calcula el discriminante: b² - 4ac | Calculates the discriminant: b² - 4ac
30 r = p - m
31
32 # Verifica si el discriminante es positivo | Checks if the discriminant is positive
33 if r > 0:
34     print('Si se puede') # La ecuación tiene dos soluciones reales | The equation has two real solutions
35
36     # Calcula la raíz cuadrada del discriminante | Calculates the square root of the discriminant
37     ra = r ** (1/2)
38
39     # Calcula 2a | Calculates 2a
40     d = 2 * a
41
42     # Calcula las dos soluciones de la ecuación | Calculates the two solutions of the equation
43     x1 = (-b + ra) / d
44     x2 = (-b - ra) / d
45
46     # Muestra los resultados con dos decimales | Displays the results with two decimals
47     print(f'El valor de x1 es: {x1:.2f} y de x2 es: {x2:.2f}')
48
49 else:
50     # Si el discriminante es negativo, no hay soluciones reales | If the discriminant is negative, no real solutions
51     print('No se puede')
52
53
54
55
56
57

```

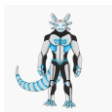
Ilustración 24 Código repaso 2

```

PS C:\Users\oscar\Dropbox\UNIVERSIDAD\Tercer s
C:\Users\oscar\AppData\Local\Microsoft\Windows
mestre/Estructura de datos/Primer semestre/Pra
Ingresa el valor de A: 10
Ingresa el valor de B: 20
Ingresa el valor de C: 4
Si se puede
El valor de x1 es: -0.23 y de x2 es: -1.77
PS C:\Users\oscar\Dropbox\UNIVERSIDAD\Tercer s

```

Ilustración 25 ejecución repaso 2





## Repaso 3

Este programa recopila los métodos que se pueden utilizar para poder realizar validaciones, también la manera de guardar información en listas

```

1
2 # def leer():
3 #     WORD que obtiene el ASCII del caracter
4 #     #ISALPHA para caracteres
5 #     #ISDIGIT para numeros
6 #     #TRY EXCEPT ValueError:
7 #     a = input('Escribe un dato o valor')
8 #     validar(a)
9
10 # def validar(a):
11 #     c=0
12 #     d=0.0
13 #     try:
14 #         c= int(a)
15 #         print('Es un valor numerico sin decimales')
16 #     except ValueError:
17 #         print('No es un valor numerico sin decimales')
18 #     try:
19 #         d = float(a)
20 #         print('Es un valor numerico con decimales')
21 #     except ValueError:
22 #         print('No es un valor numerico con decimales')
23
24 # if __name__ == '__main__':
25 #     leer()
26
27 # Hacer un programa que lea un dato, cualquiera que sea, y que lo almacene en una lista, respetando su tipo de dato
28 # Make a program that reads any input and stores it in a list, preserving its data type
29
30 list = [] # Lista para almacenar los datos ingresados | List to store the entered data
31
32 def dato(): # Función para leer un dato y validarlo | Function to read a data item and validate it
33     d = input('Ingresa un dato\n') # Solicita un dato al usuario | Asks the user for a data item
34     dato = validar(d) # Llama a la función validar para determinar el tipo de dato | Calls validar function to determine data type
35     list.append(dato) # Agrega el dato a la lista | Appends the data to the list
36
37 def validar(d): # Función para verificar el tipo de dato ingresado | Function to check the data type of the input
38     i = 0 # Variable para entero | Variable for integer
39     f = 0.0 # Variable para flotante | Variable for float
40     try:
41         i = int(d) # Intenta convertir a entero | Tries to convert to integer
42         print('Tu valor es un entero') # Mensaje si es entero | Message if input is integer
43         return i # Devuelve el entero | Returns the integer
44     except ValueError:
45         print('Tu valor no es un entero') # Mensaje si no es entero | Message if input is not integer
46     try:
47         f = float(d) # Intenta convertir a flotante | Tries to convert to float
48         print('Tu valor es un flotante') # Mensaje si es flotante | Message if input is float
49         return f # Devuelve el flotante | Returns the float
50     except ValueError:
51         print('Tu valor no es un flotante') # Mensaje si no es flotante | Message if input is not float
52
53     return d # Si no es ni entero ni flotante, devuelve el dato original | If not int or float, returns the original input
54
55 if __name__ == '__main__': # Punto de entrada del programa | Program entry point
56     while True:
57         dato() # Llama a la función dato | Calls the dato function
58         resp = input('Deseas otro S/N\n') # Pregunta si desea ingresar otro dato | Asks if user wants to enter another data item
59         if resp == 'N' or resp == 'n': # Si la respuesta es no, termina el ciclo | If the answer is no, exits the loop
60             print(list) # Muestra la lista con los datos ingresados | Displays the list with entered data
61             break
62

```

Ilustración 26 Código repaso 3

```

PS C:\Users\oscar\Dropbox\UNI
C:\Users\oscar\AppData\Local\
mestre/Estructura de datos/Pr
Ingresa un dato
40
Tu valor es un entero
Deseas otro S/N

```

Ilustración 27 ejecución repaso 3

