



# **TECNOLOGICO JOSE MARIO MOLINA PASQUEL Y HENRÍQUEZ UA ZAPOTLANEJO**

## **MANUAL DE PROGRAMADOR**

ESTRUCTURA DE DATOS

JOSE LUIS OLIVARES ACEVES



# ÍNDICE

CONTENIDO	
<b>PARCIAL 1</b>	
<a href="#">PROGRAMA 1</a>	<a href="#">3</a>
<a href="#">PROGRAMA 2</a>	<a href="#">4</a>
<a href="#">PROGRAMA 3</a>	<a href="#">5</a>
<a href="#">PROGRAMA 4</a>	<a href="#">6</a>
<a href="#">PROGRAMA 5</a>	<a href="#">7</a>
<a href="#">PROGRAMA 6</a>	<a href="#">8</a>
<a href="#">PROGRAMA 7</a>	<a href="#">9</a>
<a href="#">PROGRAMA 8</a>	<a href="#">10</a>
<a href="#">PROGRAMA 9</a>	<a href="#">11</a>
<a href="#">TAREA 1</a>	<a href="#">12</a>
<a href="#">TAREA 2</a>	<a href="#">13</a>
<a href="#">REPASO 1</a>	<a href="#">14</a>
<a href="#">REPASO 2</a>	<a href="#">15</a>
<a href="#">REPASO 3</a>	<a href="#">16</a>
<b>PARCIAL 2</b>	
<a href="#">PROGRAMA 1 P2</a>	<a href="#">17</a>
<a href="#">PROGRAMA 2 P2</a>	<a href="#">18</a>
<a href="#">PROGRAMA 3 P2</a>	<a href="#">19</a>
<a href="#">PROGRAMA 4 P2</a>	<a href="#">20</a>
<a href="#">PROGRAMA 5 P2</a>	<a href="#">21</a>
<a href="#">PROGRAMA 6 P2</a>	<a href="#">22</a>
<a href="#">PROGRAMA 7 P2</a>	<a href="#">23</a>
<a href="#">PROGRAMA REPASO P2</a>	<a href="#">24</a>
<a href="#">TAREA 1</a>	<a href="#">25</a>



# PROGRAMA 1

```
Programa1.py X
Python Programas > Programa1.py > ...
1  # int a = 0; # (ES) En C se declararía así, pero en Python no es necesario declarar el tipo | (EN) In C you would declare it like this, but in Py
2
3  a = int(input('Escribe un numero: ')) # (ES) Pide al usuario un número entero | (EN) Asks the user for an integer
4  print(a**2) # (ES) Imprime el número elevado al cuadrado | (EN) Prints the number squared
5  print(a**(1/2)) # (ES) Imprime la raíz cuadrada del número | (EN) Prints the square root of the number
6
7
8  ##operadores aritmeticos## # (ES) Lista de operadores aritméticos | (EN) List of arithmetic operators
9
10 # + # (ES) Suma | (EN) Addition
11 # - # (ES) Resta | (EN) Subtraction
12 # * # (ES) Multiplicación | (EN) Multiplication
13 # ** # (ES) Potencia | (EN) Exponentiation
14 # / # (ES) División con decimales | (EN) Division with decimals
15 # // # (ES) División entera (sin decimales) | (EN) Integer division (without decimals)
16 # % # (ES) Módulo, obtiene el residuo | (EN) Modulus, gets the remainder
17 # and, or # (ES) Operadores lógicos "y" / "o" | (EN) Logical operators "and" / "or"
18
19
20 ##Operadores diferenciales o lógicos## # (ES) Operadores de comparación y lógicos | (EN) Comparison and logical operators
21
22 # < # (ES) Menor que | (EN) Less than
23 # > # (ES) Mayor que | (EN) Greater than
24 # >= # (ES) Mayor o igual que | (EN) Greater than or equal to
25 # <= # (ES) Menor o igual que | (EN) Less than or equal to
26 # != # (ES) Diferente de | (EN) Not equal to
27 # not # (ES) Negación lógica | (EN) Logical negation
28 # == # (ES) Igual a | (EN) Equal to
29
30 # Para sacar un raíz cuadrada de un numero se deve de elevar el numero a la 1/2 # (ES) Explicación de cómo calcular raíz cuadrada | (EN) Explanat
31
```



## PROGRAMA 2

```
Programa2.py X
Python Programas > Programa2.py > ...
1  a = [10] # Español: Inicializa un arreglo con un elemento [10]. # English: Initialize an array with one element [10].
2  b = [ ] # Español: Inicializa una lista vacía. # English: Initialize an empty list.
3
4  a[0] = 10 # Español: Asigna el valor 10 al primer elemento del arreglo. # English: Assign value 10 to the first element of array.
5
6  b = {'Hola', 10, 100.05, False, 'm', {1, 2, 3, 4, 5, 6, 7}} # Español: Intenta crear un conjunto con varios tipos de datos. # English: Attempt t
7  # una lista permite guardar cualquier tipo de dato sin importa su categoria
8  # a list allows storing any type of data regardless of its category
9
10 #b.append(10) # Español: Esto está comentado, no se ejecuta. # English: This is commented, not executed.
11
12 #ciclos y condiciones
13 # cycles and conditions
14 if(len(a) > len(b)): # Español: Compara la longitud del arreglo 'a' con la del conjunto 'b'. # English: Compare length of array 'a' with set 'b'
15 |     print('A es mayor') # Español: Imprime si 'a' es mayor. # English: Print if 'a' is larger.
16
17 else: # Español: Caso contrario. # English: Otherwise.
18 |     print('B es mayor') # Español: Imprime si 'b' es mayor. # English: Print if 'b' is larger.
19
20
21 for i in a: # Español: Itera sobre cada elemento en 'a'. # English: Iterate over each element in 'a'.
22 |     print(i) # Español: Imprime el elemento actual. # English: Print the current element.
```



## PROGRAMA 3

```
1 # Hacer un progrma que lea 10 numeros y los almacene en un arreglo
2
3 a = [0,0,0,0,0,0,0,0,0,0] # Español: Inicializa un arreglo con 10 ceros. # English: Initialize an array with 10 zeros.
4
5 for i in range(0,10): # Español: Bucle que se repite 10 veces (de 0 a 9). # English: Loop that repeats 10 times (from 0 to 9).
6     a[i] = int(input(f'Ingresa un numero \n')) # Español: Solicita un número, convierte a entero y almacena en posición i. # English: Prompt for
7
8 for i in a: # Español: Itera sobre cada elemento del arreglo. # English: Iterate over each element in the array.
9     print(i) # Español: Imprime el elemento actual. # English: Print the current element.
```



## PROGRAMA 4

```
# Hacer un programa que lea 10 numeros y los almacene en una lista

a = [] # Español: Crea una lista vacía para almacenar números. # English: Create an empty list to store numbers.
s = 0 # Español: Inicializa la variable para la suma de números. # English: Initialize variable for sum of numbers.
n = 0 # Español: Inicializa el contador de números válidos. # English: Initialize counter for valid numbers.
numeros = "0,1,2,3,4,5,6,7,8,9" # Español: Define los caracteres numéricos permitidos. # English: Define allowed numeric characters.

while(n < 10): # Español: Bucle mientras no se hayan recolectado 10 números. # English: Loop until 10 numbers are collected.
    b = input('Escribe un numero: ') # Español: Solicita al usuario un número. # English: Prompt user for a number.
    x = 0 # Español: Inicializa contador de dígitos válidos. # English: Initialize counter for valid digits.
    for i in b: # Español: Itera sobre cada carácter de la entrada. # English: Iterate over each character in input.
        #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
        if i in numeros: # Español: Verifica si el carácter es numérico. # English: Check if character is numeric.
            x += 1 # Español: Incrementa contador de dígitos válidos. # English: Increment valid digit counter.
    if len(b) == x: # Español: Verifica si todos los caracteres son numéricos. # English: Check if all characters are numeric.
        a.append(int(b)) # Español: Convierte a entero y añade a la lista. # English: Convert to integer and add to list.
        n += 1 # Español: Incrementa contador de números válidos. # English: Increment valid number counter.
    else: # Español: Maneja entrada no válida. # English: Handle invalid input.
        print('El valor no es numero') # Español: Mensaje de error. # English: Error message.

for i in a: # Español: Itera sobre cada número en la lista. # English: Iterate over each number in list.
    print(i) # Español: Imprime el número. # English: Print the number.
    s += i # Español: Suma el número al total. # English: Add number to total.

print(f'La suma es: {s}') # Español: Muestra la suma total. # English: Display total sum.
```

```
Escribe un numero: 1
Escribe un numero: 2
Escribe un numero: 3
Escribe un numero: 4
Escribe un numero: 5
Escribe un numero: 6
Escribe un numero: 7
Escribe un numero: 8
Escribe un numero: 9
Escribe un numero: 10
1
2
3
4
5
6
7
8
9
10
La suma es: 55
```



## PROGRAMA 5

```

...
Hacer un programa que lea 10 datos, si el dato es un numero se almacenara en un arreglo,
si es un caracter o caracteres se almacenara en una lista,
cuando finalice el programa nos mostrara cuantos numericos y cuantos caracteres hay en cada estructura
...

ar = [-1,-1,-1,-1,-1,-1,-1,-1,-1,-1] # Español: Inicializa un arreglo con 10 elementos establecidos en -1. # English: Initialize an array
lis = [] # Español: Crea una lista vacía para valores de cadena. # English: Create an empty list for string values.
c = 0 # Español: Inicializa el contador para entradas totales. # English: Initialize the counter for total inputs.
c2 = 0 # Español: Inicializa el contador para números válidos en el arreglo. # English: Initialize the counter for valid numbers in the array.

while(True): # Español: Inicia un bucle infinito. # English: Start an infinite loop.
    a = input('Escribe un dato o valor ') # Español: Solicita al usuario un valor. # English: Prompt the user for input.
    if a.isdigit(): # Español: Verifica si la entrada es numérica. # English: Check if input is numeric.
        ar[c] = int(a) # Español: Convierte a entero y almacena en el arreglo. # English: Convert to integer and store in array.
    elif a.isalpha(): # Español: Verifica si la entrada es alfabética. # English: Check if input is alphabetic.
        lis.append(a) # Español: Agrega la cadena a la lista. # English: Add string to list.
    c += 1 # Español: Incrementa el contador de entradas. # English: Increment input counter.
    if c >= 10: # Español: Verifica si se han recolectado 10 entradas. # English: Check if 10 inputs have been collected.
        break # Español: Sale del bucle. # English: Exit loop.

print(f'La lista tiene {len(lis)}') # Español: Muestra la cantidad de elementos de cadena. # English: Display count of string elements

for i in ar: # Español: Itera a través del arreglo. # English: Iterate through array.
    if i != -1: # Español: Verifica número válido (no -1). # English: Check for valid number (not -1).
        c2 += 1 # Español: Cuenta números válidos. # English: Count valid numbers.

print(f'El arreglo tiene {c2}') # Español: Muestra la cantidad de elementos numéricos. # English: Display count of numeric elements.
print(ar) # Español: Imprime el arreglo. # English: Print the array.
print(lis) # Español: Imprime la lista. # English: Print the list.

...

for i in range(0,10):
    a = input('Escribe un numero: ')
    if a.isdigit():
        ar[i]=a
    if a.isalpha():
        lis.append(a)

```

```

Escribe un dato o valor 1
Escribe un dato o valor 2
Escribe un dato o valor 3
Escribe un dato o valor 4
Escribe un dato o valor 5
Escribe un dato o valor a
Escribe un dato o valor e
Escribe un dato o valor i
Escribe un dato o valor o
Escribe un dato o valor u
La lista tiene 5
El arreglo tiene 5
[1, 2, 3, 4, 5, -1, -1, -1, -1, -1]
['a', 'e', 'i', 'o', 'u']

```



## PROGRAMA 6

```
def resultados():
    c2 = 0 # Español: Inicializa el contador c2 en 0. # English: Initialize the counter c2 to 0.
    print(f'La lista tiene {len(lis)}') # Español: Imprime la longitud de la lista 'lis'. # English: Print the length of the list 'lis'.
    for i in ar: # Español: Recorre cada elemento del arreglo 'ar'. # English: Iterate over each element of array 'ar'.
        if i != -1: # Español: Comprueba si el elemento no es -1 (valor marcador). # English: Check if the element is not -1 (marker value).
            c2 += 1 # Español: Si no es -1, incrementa el contador c2. # English: If it's not -1, increment counter c2.
    print(f'El arreglo tiene {c2}') # Español: Imprime cuántos elementos distintos de -1 hay en 'ar'. # English: Print how many elements
    print(ar) # Español: Imprime el contenido completo del arreglo 'ar'. # English: Print the full contents of the array 'ar'.
    print(lis) # Español: Imprime el contenido de la lista 'lis'. # English: Print the contents of the list 'lis'.

def hola(): # definicion de metodo o funcion
    # Español: Define la función 'hola' que leerá hasta 10 entradas y las catalogará en 'ar' o 'lis'.
    # English: Define the 'hola' function that will read up to 10 inputs and categorize them into 'ar' or 'lis'.
    c = 0 # Español: Inicializa el índice/contador 'c' en 0 para usar como posición en 'ar'. # English: Initialize index/counter 'c' to 0.
    while(True): # Español: Bucle que leerá datos repetidamente hasta alcanzar 10 entradas. # English: Loop that will read data repeatedly
        a = input('Escribe un dato o valor ') # Español: Pide al usuario un dato y lo guarda en 'a' como cadena. # English: Ask the user for input
        if a.isdigit(): # Español: Si la entrada contiene solo dígitos (número positivo sin signo). # English: If the input contains only digits
            ar[c] = int(a) # Español: Convierte a entero y lo almacena en la posición c del arreglo 'ar'. # English: Convert to int and store in array
        elif a.isalpha(): # Español: Si la entrada contiene solo letras. # English: If the input contains only letters
            lis.append(a) # Español: Añade la cadena a la lista 'lis'. # English: Append the string to the list 'lis'.
        c += 1 # Español: Incrementa el índice/contador para la siguiente posición del arreglo. # English: Increment the index/counter for next position
        if c >= 10: # Español: Si se han leído 10 entradas, sale del bucle. # English: If 10 entries have been read, exit the loop.
            break # Español: Rompe el while. # English: Break the while loop.
    resultados() # Español: Llama a la función 'resultados' para mostrar salidas. # English: Call the 'resultados' function to display outputs

ar = [-1,-1,-1,-1,-1,-1,-1,-1,-1,-1] # Español: Arreglo con 10 posiciones inicializadas a -1 como marcador. # English: Array with 10 positions initialized to -1 as marker.
lis = [] # Español: Lista vacía donde se almacenarán entradas alfabéticas. # English: Empty list where alphabetic entries will be stored

if __name__ == "__main__": # Español: Comprueba si el archivo se ejecuta directamente. # English: Check if the file is executed directly
    hola() # Español: Si la condición anterior se cumple, llama a 'hola()'. # English: If the above condition holds, call 'hola()'.
```





# PROGRAMA 7

```
# Hacer un programa que lea nombre, edad, y sexo de 5 personas, estos elementos tienen que estar dentro de una lista

def inicio(): # Español: Define la función 'inicio' que controla el flujo principal. # English: Define the 'inicio' function that controls the m
    c=0 # Español: Inicializa el contador 'c' en 0. # English: Initialize the counter 'c' to 0.
    while (True): # Español: Inicia un bucle infinito que repetirá la lectura hasta alcanzar 5 registros. # English: Start an infinite loop that
        nombre() # Español: Llama a la función 'nombre' para leer y validar un nombre. # English: Call the 'nombre' function to read and validat
        edad() # Español: Llama a la función 'edad' para leer y validar una edad. # English: Call the 'edad' function to read and validate an ag
        sexo() # Español: Llama a la función 'sexo' para leer y validar el sexo (M/F). # English: Call the 'sexo' function to read and validate
        c+=1 # Español: Incrementa el contador 'c' después de leer un registro completo. # English: Increment the counter 'c' after reading a co
        if c >= 5: # Español: Comprueba si se han leído 5 o más registros. # English: Check if 5 or more records have been read.
            break # Español: Si el contador llega a 5, rompe el bucle y termina la función. # English: If the counter reaches 5, break the loop

def nombre(): # Español: Define la función 'nombre' para pedir y validar un nombre. # English: Define the 'nombre' function to request and valid
    while(True): # Español: Bucle que se repetirá hasta recibir un nombre válido. # English: Loop that repeats until a valid name is received.
        n=input('Escribe un nombre ') # Español: Pide al usuario que escriba un nombre y lo guarda en 'n'. # English: Ask the user to write a na
        if n.isalpha(): # Español: Verifica que 'n' solo contenga letras (sin espacios ni dígitos). # English: Verify that 'n' contains only let
            lis.append(n) # Español: Si es válido, añade el nombre a la lista 'lis'. # English: If valid, append the name to the list 'lis'.
            break # Español: Sale del bucle una vez añadido el nombre. # English: Exit the loop once the name has been appended.
        else:
            print('Nombre no valido') # Español: Mensaje si el nombre no es válido. # English: Message shown if the name is not valid.

def edad(): # Español: Define la función 'edad' para pedir y validar la edad. # English: Define the 'edad' function to request and validate age.
    while(True): # Español: Bucle que se repetirá hasta recibir una edad válida. # English: Loop that repeats until a valid age is received.
        e=input('Escribe una edad ') # Español: Pide al usuario que escriba una edad y la guarda en 'e'. # English: Ask the user to write an age
        if e.isdigit(): # Español: Verifica que 'e' esté compuesta solo por dígitos. # English: Verify that 'e' is composed only of digits.
            lis.append(e) # Español: Si es válido, añade la edad a la lista 'lis'. # English: If valid, append the age to the list 'lis'.
            break # Español: Sale del bucle una vez añadida la edad. # English: Exit the loop once the age has been appended.
        else:
            print('Edad no valida') # Español: Mensaje si la edad no es válida. # English: Message shown if the age is not valid.

def sexo(): # Español: Define la función 'sexo' para pedir y validar el sexo (M/F). # English: Define the 'sexo' function to request and validat
    while(True): # Español: Bucle que se repetirá hasta recibir un sexo válido. # English: Loop that repeats until a valid sex is received.
        s=input('Escribe un sexo (M/F) ') # Español: Pide al usuario que indique el sexo y lo guarda en 's'. # English: Ask the user to indicate
        if s=='M' or s=='F': # Español: Verifica que la entrada sea exactamente 'M' o 'F'. # English: Verify that the input is exactly 'M' or 'F'.
            lis.append(s) # Español: Si es válido, añade el sexo a la lista 'lis'. # English: If valid, append the sex to the list 'lis'.
            break # Español: Sale del bucle una vez añadido el sexo. # English: Exit the loop once the sex has been appended.
        else:
            print('Sexo no valido') # Español: Mensaje si la entrada no es 'M' ni 'F'. # English: Message shown if the input is not 'M' or 'F'.

def resultados(): # Español: Define la función 'resultados' (en el código actual imprime desde la lista). # English: Define the 'resultados' fun
    i = 0 # Español: Inicializa el índice 'i' en 0. # English: Initialize the index 'i' to 0.
    while(True): # Español: Bucle infinito que imprimirá elementos de 'lis' (tal como está, no incrementa 'i'). # English: Infinite loop that wi
        print(lis[i]) # Español: Imprime el elemento de la lista en la posición 'i'. # English: Print the element of the list at position 'i'.

lis = [] # Español: Inicializa la lista vacía 'lis' donde se almacenarán nombres, edades y sexos. # English: Initialize the empty list 'lis' whe
inicio() # Español: Llama a la función 'inicio' para comenzar el proceso de ingreso. # English: Call the 'inicio' function to start the input pr
```



## PROGRAMA 8

```
'''
Hacer un programa que lea una cadena y que muestre en pantalla cuantos numeros tiene, cuantas mayusculas, cuantas minusculas
y cuantos espacios
'''
def inicio():
    n=0 # Español: Inicializa el contador de números en 0. # English: Initialize the numbers counter to 0.
    e=0 # Español: Inicializa el contador de espacios en 0. # English: Initialize the spaces counter to 0.
    min=0 # Español: Inicializa el contador de minúsculas en 0. # English: Initialize the lowercase counter to 0.
    may=0 # Español: Inicializa el contador de mayúsculas en 0. # English: Initialize the uppercase counter to 0.
    numeros = "0123456789" # Español: Cadena con todos los dígitos usada para comprobaciones. # English: String with all digits used for checks.
    cadena = input('Escribe una cadena ') # Español: Solicita una cadena al usuario y la guarda en 'cadena'. # English: Prompt the user for a string.
    for i in cadena: # Español: Recorre cada carácter de la cadena. # English: Iterate over each character in the string.
        if i in numeros: # Español: Comprueba si el carácter es un dígito (está en 'numeros'). # English: Check if the character is a digit (is in 'numeros').
            print('Es numero') # Español: Imprime que el carácter es un número. # English: Print that the character is a number.
            n+=1 # Español: Incrementa el contador de números. # English: Increment the numbers counter.
        if i == ' ': # Español: Comprueba si el carácter es un espacio. # English: Check if the character is a space.
            e+=1 # Español: Incrementa el contador de espacios. # English: Increment the spaces counter.
        if ord(i)>=97 and ord(i)<=122: # Español: Comprueba si el carácter es una minúscula según su código ASCII. # English: Check if the character is a lowercase letter according to its ASCII code.
            min+=1 # Español: Incrementa el contador de minúsculas. # English: Increment the lowercase counter.
        if ord(i)>=65 and ord(i)<=90: # Español: Comprueba si el carácter es una mayúscula según su código ASCII. # English: Check if the character is an uppercase letter according to its ASCII code.
            may+=1 # Español: Incrementa el contador de mayúsculas. # English: Increment the uppercase counter.
    print(f'Los numeros son: {n} \nLos espacios son: {e} \nLas minusculas son: {min} \nLas mayusculas son: {may}') # Español: Muestra en pantalla los resultados. # English: Display the results on the screen.

if __name__ == '__main__':
    inicio() # Español: Llama a la función inicio para ejecutar el programa si el archivo se ejecuta directamente. # English: Call the inicio function to execute the program if the file is executed directly.
```

```
Escribe una cadena Hola mundo
Los numeros son: 0
Los espacios son: 1
Las minusculas son: 8
Las mayusculas son: 1
PS C:\Users\iosel\Documents\Py
```



## PROGRAMA 9

```

1  """
2  Hacer un programa que en una lista se introduzca cadenas de caracteres con las siguientes restricciones
3  1 - las cadenas no deben de haber espacios
4  2- la cadena solo puede tener mayuscula la primer letra
5  3- obligatoriamente debe de tener todas las vocales
6  el programa no termina hasta que la lista tenga 5 elementos
7  """
8  def vocales(cad):
9      ba = False # Español: Inicializa bandera para la vocal 'a'. # English: Initialize flag for vowel 'a'.
10     be = False # Español: Inicializa bandera para la vocal 'e'. # English: Initialize flag for vowel 'e'.
11     bi = False # Español: Inicializa bandera para la vocal 'i'. # English: Initialize flag for vowel 'i'.
12     bo = False # Español: Inicializa bandera para la vocal 'o'. # English: Initialize flag for vowel 'o'.
13     bu = False # Español: Inicializa bandera para la vocal 'u'. # English: Initialize flag for vowel 'u'.
14     if 'a' in cad or 'A' in cad:
15         ba = True # Español: Si 'a' o 'A' está en la cadena, marca ba True. # English: If 'a' or 'A' is in the string, set ba True.
16     if 'e' in cad or 'E' in cad:
17         be = True # Español: Si 'e' o 'E' está en la cadena, marca be True. # English: If 'e' or 'E' is in the string, set be True.
18     if 'i' in cad or 'I' in cad:
19         bi = True # Español: Si 'i' o 'I' está en la cadena, marca bi True. # English: If 'i' or 'I' is in the string, set bi True.
20     if 'o' in cad or 'O' in cad:
21         bo = True # Español: Si 'o' o 'O' está en la cadena, marca bo True. # English: If 'o' or 'O' is in the string, set bo True.
22     if 'u' in cad or 'U' in cad:
23         bu = True # Español: Si 'u' o 'U' está en la cadena, marca bu True. # English: If 'u' or 'U' is in the string, set bu True.
24     if ba==True and be==True and bi==True and bo==True and bu==True:
25         lista.append(cad) # Español: Si todas las vocales están, añade la cadena a la lista. # English: If all vowels are present, append the string to the list.
26     print(lista) # Español: Muestra el contenido actual de la lista. # English: Show the current content of the list.
27
28 def minusculas(c1):
29     cm = 0 # Español: Contador de minúsculas (excepto la primera) inicializado a 0. # English: Counter for lowercase letters (except first) initialized to 0.
30     print(c1) # Español: Imprime la cadena recibida. # English: Print the received string.
31     for i in c1[1:]:
32         if ord(i) >= 97 and ord(i) <=122:
33             cm += 1 # Español: Si el caracter es letra minúscula (ascii 97-122), incrementa el contador. # English: If the character is lowercase, increment the counter.
34     if cm == len(c1)-1 :
35         print(f'La cadena son minusculas excepto la primera {cm}') # Español: Indica que todas las letras (excepto la primera) son minúsculas. # English: Indicate that all letters (except the first) are lowercase.
36         vocales(cm) # Español: Llama a la función 'vocales' pasando 'cm' (nota: en el código original se pasa un entero). # English: Call 'vocales' with 'cm'.
37     else:
38         print('Error la cadena no cumple') # Español: Indica que la cadena no cumple la restricción de mayúsculas/minúsculas. # English: Indicate that the string does not meet the uppercase/lowercase restriction.
39
40 def inicio():
41     ncadena = "" # Español: Inicializa una cadena vacía que se usará para formar una versión sin caracteres no alfabéticos. # English: Initialize an empty string that will be used to form a version without non-alphabetic characters.
42     cadena = input('Escribe una cadena ') # Español: Pide al usuario que escriba una cadena y la guarda en 'cadena'. # English: Ask the user to enter a string and save it in 'cadena'.
43     for i in cadena:
44         if ord(i) != 32: # validar sin espacios
45             if cadena.isalpha():
46                 minusculas(cadena) # Español: Si toda la cadena es alfabética, llama a 'minusculas' con la cadena original. # English: If the string is alphabetic, call 'minusculas' with the original string.
47             else:
48                 for i in cadena:
49                     if ord(i)>=48 and ord(i)>=57:
50                         pass # Español: Si el caracter cumple la condición (según el código original), no hace nada. # English: If the character meets the condition, do nothing.
51                     else:
52                         ncadena+= i # Español: Si no, concatena el caracter en 'ncadena'. # English: Otherwise, concatenate the character into 'ncadena'.
53                 minusculas(ncadena) # Español: Llama a 'minusculas' con la cadena filtrada 'ncadena'. # English: Call 'minusculas' with the filtered string.
54
55 lista = [] # Español: Inicializa la lista vacía donde se almacenarán las cadenas válidas. # English: Initialize the empty list where valid strings will be stored.
56 if __name__ == '__main__':
57     while(True):
58         inicio() # Español: Llama a 'inicio' repetidamente hasta que la lista alcance 5 elementos. # English: Call 'inicio' repeatedly until the list reaches 5 elements.
59         if len(lista) >= 5:
60             break # Español: Si la lista tiene 5 o más elementos, sale del bucle y termina el programa. # English: If the list has 5 or more elements, exit the loop and end the program.

```



# TAREA 1

```
'''
Hacer un programa que lea 10 datos, si el dato es un numero se almacenara en un arreglo,
si es un caracter o caracteres se almacenara en una lista,
cuando finalice el programa nos mostrara cuantos numericos y cuantos caracteres hay en cada estructura
'''

arreglo = [0,0,0,0,0,0,0,0,0,0] # Español: Inicializa un arreglo con 10 ceros. # English: Initialize an array with 10 zeros.
lista = [] # Español: Inicializa una lista vacía. # English: Initialize an empty list.
cn = 0 # Español: Inicializa contador para números. # English: Initialize counter for numbers.
cc = 0 # Español: Inicializa contador para caracteres. # English: Initialize counter for characters.

for i in range(0,10): # Español: Bucle que se repite 10 veces. # English: Loop that repeats 10 times.

    d = input('Escribe un dato: ') # Español: Solicita al usuario un dato. # English: Prompt user for data.
    if d.isdigit(): # Español: Verifica si el dato es numérico. # English: Check if data is numeric.
        arreglo[i] = d # Español: Almacena el número en el arreglo. # English: Store number in array.
        cn += 1 # Español: Incrementa contador de números. # English: Increment number counter.
    elif d.isalpha(): # Español: Verifica si el dato es alfabético. # English: Check if data is alphabetic.
        lista.append(d) # Español: Agrega caracteres a la lista. # English: Add characters to list.
        cc += 1 # Español: Incrementa contador de caracteres. # English: Increment character counter.

print(f'El arreglo tiene: {cn}\nLa lista tiene: {cc}') # Español: Muestra cantidad de números y caracteres. # English: Show count of numbers and
print(arreglo) # Español: Imprime el arreglo. # English: Print the array.
print(lista) # Español: Imprime la lista. # English: Print the list.
```

```
PS C:\Users\josel\Documents\Python Progra
Escribe un dato: 1
Escribe un dato: 2
Escribe un dato: 3
Escribe un dato: 4
Escribe un dato: 5
Escribe un dato: a
Escribe un dato: e
Escribe un dato: ii
Escribe un dato: o
Escribe un dato: u
El arreglo tiene: 5
La lista tiene: 5
['1', '2', '3', '4', '5', 0, 0, 0, 0, 0]
['a', 'e', 'ii', 'o', 'u']
```



## TAREA 2

```
'''
Hacer un programa que en una lista se introduzca cadenas de caracteres con las siguientes restricciones
1 - las cadenas no deben de haber espacios
2- la cadena solo puede tener mayuscula la primer letra
3- obligatoriamente debe de tener todas las vocales
el programa no termina hasta que la lista tenga 5 elementos
'''

def vocales(cad): # Español: Define función para verificar si contiene todas las vocales. # English: Define function to check if contains all
    b = False # Español: Inicializa bandera como falsa. # English: Initialize flag as false.
    if 'a' in cad or 'A' in cad: # Español: Verifica si contiene 'a' o 'A'. # English: Check if contains 'a' or 'A'.
        if 'e' in cad or 'E' in cad: # Español: Verifica si contiene 'e' o 'E'. # English: Check if contains 'e' or 'E'.
            if 'i' in cad or 'I' in cad: # Español: Verifica si contiene 'i' o 'I'. # English: Check if contains 'i' or 'I'.
                if 'o' in cad or 'O' in cad: # Español: Verifica si contiene 'o' o 'O'. # English: Check if contains 'o' or 'O'.
                    if 'u' in cad or 'U' in cad: # Español: Verifica si contiene 'u' o 'U'. # English: Check if contains 'u' or 'U'.
                        b = True # Español: Establece bandera como verdadera. # English: Set flag as true.
                        return b # Español: Retorna verdadero. # English: Return true.
                    else: # Español: Si no tiene 'u' o 'U'. # English: If doesn't have 'u' or 'U'.
                        return b # Español: Retorna falso. # English: Return false.
                else: # Español: Si no tiene 'o' o 'O'. # English: If doesn't have 'o' or 'O'.
                    return b # Español: Retorna falso. # English: Return false.
            else: # Español: Si no tiene 'i' o 'I'. # English: If doesn't have 'i' or 'I'.
                return b # Español: Retorna falso. # English: Return false.
        else: # Español: Si no tiene 'e' o 'E'. # English: If doesn't have 'e' or 'E'.
            return b # Español: Retorna falso. # English: Return false.
    else: # Español: Si no tiene 'a' o 'A'. # English: If doesn't have 'a' or 'A'.
        return b # Español: Retorna falso. # English: Return false.

def minusculas(cad): # Español: Define función para verificar minúsculas después del primer carácter. # English: Define function to check lower
    a = False # Español: Inicializa bandera como falsa. # English: Initialize flag as false.
    for i in cad[1:]: # Español: Itera desde el segundo carácter hasta el final. # English: Iterate from second character to end.
        if ord(i) >= 97 and ord(i) <= 122: # Español: Verifica si es minúscula (código ASCII). # English: Check if lowercase (ASCII code).
            pass # Español: No hace nada si es minúscula. # English: Do nothing if lowercase.
        else: # Español: Si no es minúscula. # English: If not lowercase.
            return a # Español: Retorna falso inmediatamente. # English: Return false immediately.
    a = True # Español: Si todos los caracteres son minúsculas. # English: If all characters are lowercase.
    return a # Español: Retorna verdadero. # English: Return true.

def inicio(): # Español: Función principal que solicita y valida la entrada. # English: Main function that requests and validates input.
    cad = input('Escribe una cadena: ') # Español: Solicita una cadena al usuario. # English: Prompt user for a string.

    rvoc = vocales(cad) # Español: Verifica si tiene todas las vocales. # English: Check if has all vowels.
    if rvoc == True: # Español: Si tiene todas las vocales. # English: If has all vowels.
        if not ' ' in cad: # Español: Verifica que no tenga espacios. # English: Check that it has no spaces.
            rmin = minusculas(cad) # Español: Verifica minúsculas después del primer carácter. # English: Check lowercase after first character.
            if rmin == True: # Español: Si cumple regla de minúsculas. # English: If follows lowercase rule.
                lista.append(cad) # Español: Agrega cadena válida a la lista. # English: Add valid string to list.
            else: # Español: Si no cumple regla de minúsculas. # English: If doesn't follow lowercase rule.
                print('Solo puede ser mayuscula el primer caracter') # Español: Mensaje de error. # English: Error message.
        else: # Español: Si contiene espacios. # English: If contains spaces.
            print('No debe de tener espacios') # Español: Mensaje de error. # English: Error message.
    else: # Español: Si no tiene todas las vocales. # English: If doesn't have all vowels.
        print('Necesita tener todas las vocales') # Español: Mensaje de error. # English: Error message.

lista = [] # Español: Inicializa lista vacía. # English: Initialize empty list.
if __name__ == '__main__': # Español: Punto de entrada del programa. # English: Program entry point.
    while(True): # Español: Bucle infinito. # English: Infinite loop.
        inicio() # Español: Llama a función inicio. # English: Call inicio function.
        if len(lista) >= 5: # Español: Verifica si la lista tiene 5 elementos. # English: Check if list has 5 elements.
            print(lista) # Español: Imprime la lista. # English: Print the list.
            break # Español: Termina el bucle. # English: End loop.
```

Escribe una cadena: Hola mundo

Necesita tener todas las vocales

Escribe una cadena: aeiou

Escribe una cadena: aEIOU

Solo puede ser mayuscula el primer caracter

Escribe una cadena: a e iou

No debe de tener espacios

Escribe una cadena: Eaioiu

Escribe una cadena: aeioiu

Escribe una cadena: aeioiu

Escribe una cadena: aeioiu

['aeiou', 'Eaioiu', 'aeioiu', 'aeioiu', 'aeioiu']



# REPASO 1

```
# Instrucciones de entrada y salida
# print() o print(f)
#print('Hola Mundo')
#print(f'Hola mundo numeros {10}')
# Entrada de datos
#input('Escribe un numero') # se introducen solo letras
# casting para convertir a valores especificos
#f = 0.0
#f = float(input('Escribe un numero con decimales'))
#a = 0
#a = int(input('Escribe un numero con decimales'))
#c = 120
#print(str(c))
#v = ""
#v = str(c)
#NOTA: solo las variables que no se introducen por teclado se obliga a inicializarlas

...

Hacer un programa que lea nombre y precio de un producto, el programa calculara el costo y precio de venta
El costo involucra el 12% y IVA 16%
...

# for i in range(1,5): #rango valor inicial hasta valor final sin incluirlo

while(True): # Español: Inicia un bucle infinito que repetirá el bloque hasta encontrar un break. # English: Start an infinite loop that repeats
    n = input('Escribe el nombre de tu producto: ') # Español: Lee el nombre del producto como cadena y lo guarda en `n`. # English: Read the product name
    p = float(input('Escribe el precio del producto: ')) # Español: Lee el precio (texto) y lo convierte a float, luego lo asigna a `p`. # English: Read the price
    c = p * 1.12 # Español: Calcula el costo aplicando un 12% (c = p * 1.12). # English: Calculate the cost applying 12% (c = p * 1.12).
    pv = c * 1.16 # Español: Calcula el precio de venta aplicando IVA del 16% sobre el costo (pv = c * 1.16). # English: Calculate the selling price applying 16%
    print(f'El costo es: {c:.2f}') # Español: Imprime el costo con 2 decimales. # English: Print the cost with 2 decimal places.
    print(f'El precio de venta es: {pv:.2f}') # Español: Imprime el precio de venta con 2 decimales. # English: Print the selling price with 2 decimal places.

    res = input('Deseas otro numero (s/n) \n') # Español: Pregunta si desea procesar otro producto; guarda la respuesta en `res`. # English: Ask if you want another number
    if res == 'n' or res == 'N': # Español: Si la respuesta es 'n' o 'N', se sale del bucle. # English: If the response is 'n' or 'N', exit the loop
        break # Español: Rompe el while y termina el programa. # English: Break the while and end the program.
```

```
Escribe el nombre de tu producto: Tomate
Escribe el precio del producto: 10
El costo es: 11.20
El precio de venta es: 12.99
Deseas otro numero (s/n)
```



## REPASO 2

```

1
2
3
4
5 a = 1
6 # Español: Asigna el valor entero 1 a la variable 'a' (coeficiente cuadrático).
7 # English: Assign the integer 1 to variable 'a' (quadratic coefficient).
8
9 b = 2
10 # Español: Asigna el valor entero 2 a la variable 'b' (coeficiente lineal).
11 # English: Assign the integer 2 to variable 'b' (linear coefficient).
12
13 c = -15
14 # Español: Asigna el valor entero -15 a la variable 'c' (término independiente).
15 # English: Assign the integer -15 to variable 'c' (constant term).
16
17 p = 0
18 # Español: Inicializa la variable 'p' con 0; aquí se usará para b**2.
19 # English: Initialize variable 'p' to 0; here it will be used for b**2.
20
21 m = 0
22 # Español: Inicializa la variable 'm' con 0; aquí se usará para 4*a*c.
23 # English: Initialize variable 'm' to 0; here it will be used for 4*a*c.
24
25 r = 0
26 # Español: Inicializa la variable 'r' con 0; representará el discriminante (b^2 - 4ac).
27 # English: Initialize variable 'r' to 0; it will represent the discriminant (b^2 - 4ac).
28
29 ra = 0.0
30 # Español: Inicializa 'ra' como float; se usará para la raíz cuadrada del discriminante.
31 # English: Initialize 'ra' as float; it will be used for the square root of the discriminant.
32
33 d = 0.0
34 # Español: Inicializa 'd' como float; se usará para el denominador 2*a.
35 # English: Initialize 'd' as float; it will be used for the denominator 2*a.
36
37 x1 = 0.0
38 # Español: Inicializa 'x1' como float; contendrá la primera raíz.
39 # English: Initialize 'x1' as float; it will hold the first root.
40
41 x2 = 0.0
42 # Español: Inicializa 'x2' como float; contendrá la segunda raíz.
43 # English: Initialize 'x2' as float; it will hold the second root.
44
45
46
47 p = b**2
48 # Español: Calcula b al cuadrado y lo asigna a 'p'.
49 # English: Compute b squared and assign it to 'p'.
50
51 m = 4*a*c
52 # Español: Calcula 4*a*c y lo asigna a 'm'.
53 # English: Compute 4*a*c and assign it to 'm'.
54
55 r = p - m
56 # Español: Calcula el discriminante r = b^2 - 4ac.
57 # English: Compute the discriminant r = b^2 - 4ac.
58
59 if r > 0:
60     # Español: Si el discriminante es mayor que 0 hay dos raíces reales distintas.
61     # English: If the discriminant is greater than 0 there are two distinct real roots.
62     print('Si se puede')
63     # Español: Imprime mensaje indicando que sí es posible calcular raíces reales.
64     # English: Print a message indicating it's possible to compute real roots.
65     ra = r ** (1/2)
66     # Español: Calcula la raíz cuadrada del discriminante y la guarda en 'ra'.
67     # English: Calculate the square root of the discriminant and store it in 'ra'.
68     d = 2*a
69     # Español: Calcula el denominador 2*a y lo guarda en 'd'.
70     # English: Compute the denominator 2*a and store it in 'd'.
71     x1 = (-b + ra) / d
72     # Español: Calcula la primera raíz usando la fórmula (-b + sqrt(r)) / (2a).
73     # English: Calculate the first root using the formula (-b + sqrt(r)) / (2a).
74     x2 = (-b - ra) / d
75     # Español: Calcula la segunda raíz usando la fórmula (-b - sqrt(r)) / (2a).
76     # English: Calculate the second root using the formula (-b - sqrt(r)) / (2a).
77     print(f'El valor de x1 es {x1:.2f} y de x2 es {x2:.2f}')
78     # Español: Imprime los valores de x1 y x2 formateados con 2 decimales.
79     # English: Print the values of x1 and x2 formatted with 2 decimal places.
80 else:
81     # Español: Si r no es mayor que 0 (es 0 o negativo) entra aquí.
82     # English: If r is not greater than 0 (it is 0 or negative) enter here.
83     print('No se puede')
84     # Español: Imprime mensaje indicando que no es posible calcular dos raíces reales distintas.

```

Si se puede  
El valor de x1 es 3.00 y de x2 es -5.00





## REPASO 3

```

1 def validar(a):
2     # Español: Define la función 'validar' que recibe el parámetro 'a'.
3     # English: Define the function 'validar' that receives the parameter 'a'.
4     c = 0
5     # Español: Inicializa la variable entera 'c' con 0.
6     # English: Initialize the integer variable 'c' to 0.
7     d = 0.0
8     # Español: Inicializa la variable decimal 'd' con 0.0.
9     # English: Initialize the float variable 'd' to 0.0.
10    try:
11        # Español: Intenta ejecutar el bloque que sigue (intentar conversión a entero).
12        # English: Try to execute the following block (attempt conversion to int).
13        c = int(a)
14        # Español: Convierte 'a' a entero y lo asigna a 'c'.
15        # English: Convert 'a' to an integer and assign it to 'c'.
16        print('Es un valor numerico sin decimales')
17        # Español: Muestra mensaje indicando que es un número sin decimales.
18        # English: Print message indicating it is a number without decimals.
19    except ValueError:
20        # Español: Si ocurre un ValueError (no se puede convertir a int), entra aquí.
21        # English: If a ValueError occurs (cannot convert to int), enter here.
22        print('No es un valor numerico sin decimales')
23        # Español: Muestra mensaje indicando que no es un número entero.
24        # English: Print message indicating it is not an integer number.
25
26    try:
27        # Español: Intenta ejecutar el bloque que sigue (intentar conversión a float).
28        # English: Try to execute the following block (attempt conversion to float).
29        d = float(a)
30        # Español: Convierte 'a' a float y lo asigna a 'd'.
31        # English: Convert 'a' to a float and assign it to 'd'.
32        print('Es un valor numerico con decimales')
33        # Español: Muestra mensaje indicando que es un número con decimales.
34        # English: Print message indicating it is a number with decimals.
35    except ValueError:
36        # Español: Si ocurre un ValueError (no se puede convertir a float), entra aquí.
37        # English: If a ValueError occurs (cannot convert to float), enter here.
38        print('No es un valor numerico con decimales')
39        # Español: Muestra mensaje indicando que no es un número con decimales.
40        # English: Print message indicating it is not a number with decimals.
41
42 def listas(d):
43     # Español: Define la función 'listas' que intenta reconocer el tipo de dato y devolverlo en su tipo.
44     # English: Define the function 'listas' that tries to recognize the data type and return it in its type.
45     en = 0
46     # Español: Inicializa la variable entera 'en' con 0.
47     # English: Initialize the integer variable 'en' to 0.
48     dec = 0.0
49     # Español: Inicializa la variable decimal 'dec' con 0.0.
50     # English: Initialize the float variable 'dec' to 0.0.
51
52     try:
53         # Español: Intenta convertir 'd' a entero.
54         # English: Try to convert 'd' to an integer.
55         en = int(d)
56         # Español: Asigna el entero convertido a 'en'.
57         # English: Assign the converted integer to 'en'.
58         return en
59         # Español: Devuelve el entero si la conversión fue exitosa.
60         # English: Return the integer if conversion succeeded.
61     except ValueError:
62         # Español: Si falla la conversión a entero, se captura aquí.
63         # English: If integer conversion fails, it is caught here.
64         print('No es un entero')
65         # Español: Informa que no es un entero.
66         # English: Inform that it is not an integer.
67
68     try:
69         # Español: Intenta convertir 'd' a float.
70         # English: Try to convert 'd' to a float.
71         dec = float(d)
72         # Español: Asigna el float convertido a 'dec'.
73         # English: Assign the converted float to 'dec'.
74         return dec
75         # Español: Devuelve el float si la conversión fue exitosa.
76         # English: Return the float if conversion succeeded.
77     except ValueError:
78         # Español: Si falla la conversión a float, se captura aquí.
79         # English: If float conversion fails, it is caught here.
80         print('No es un numero con decimales')
81         # Español: Informa que no es un número con decimales.
82         # English: Inform that it is not a number with decimals.
83
84     return d
85     # Español: Si no es ni entero ni float, devuelve el valor original (como string).
86     # English: If it's neither int nor float, return the original value (as string).
87
88 def leer():
89     # Español: Define la función 'leer' que solicita un dato al usuario y lo procesa.
90     # English: Define the function 'leer' that requests data from the user and processes it.
91     d = input('Escribe un dato: ')
92     # Español: Lee una entrada del usuario y la guarda en 'd'.
93     # English: Read user input and store it in 'd'.
94     dato = listas(d)
95     # Español: Llama a 'listas' para convertir 'd' a su tipo si es posible; guarda el resultado en 'dato'.
96     # English: Call 'listas' to convert 'd' to its type if possible; store the result in 'dato'.
97     lista.append(dato)
98     # Español: Añade el dato (convertido o no) a la lista global 'lista'.
99     # English: Append the data (converted or not) to the global list 'lista'.
100
101 # Español: Inicializa la lista vacía global donde se almacenarán los datos.
102 # English: Initialize the empty global list where data will be stored.
103 lista = []
104
105 if __name__ == '__main__':
106     # Español: Comprueba si el script está siendo ejecutado directamente (no importado).
107     # English: Check if the script is being run directly (not imported).
108     while True:
109         # Español: Bucle infinito que permite leer repetidamente datos del usuario.
110         # English: Infinite loop that allows repeatedly reading user data.
111         leer()
112         # Español: Llama a la función 'leer' para solicitar y almacenar un dato.
113         # English: Call the 'leer' function to request and store one datum.
114
115         res = input('Deseas otro dato (S/N): ')
116         # Español: Pregunta al usuario si desea ingresar otro dato y guarda la respuesta en 'res'.
117         # English: Ask the user if they want to enter another datum and save the response in 'res'.
118         if res == 'n' or res == 'N':
119             # Español: Si la respuesta es 'n' o 'N', termina el bucle.
120             # English: If the response is 'n' or 'N', break the loop.
121             break
122         # Español: Imprime la lista con todos los datos almacenados.
123         # English: Print the list with all stored data.
124         print(lista)
125     # Español: Rompe el bucle 'while' terminando la ejecución principal.
126     # English: Break the 'while' loop terminating the main execution.

```





# PARCIAL 2

## PROGRAMA 1 P2

```
Practicas Parcial 2 > Program1.py
1 def inicio(num): # Español: Se define una función llamada 'inicio' que recibe un parámetro 'num' / English: Defines a function called 'inicio' th
2     a = int(input("Escribe una calificacion \n")) # Español: Pide al usuario una calificación y la convierte a entero / English: Asks the user fo
3     num += 1 # Español: Incrementa el valor de 'num' en 1 / English: Increases the value of 'num' by 1
4     lista.append(a) # Español: Agrega la calificación a la lista 'lista' / English: Adds the grade to the list 'lista'
5     if num >= 5: # Español: Si se han ingresado 5 o más calificaciones... / English: If 5 or more grades have been entered...
6         print(lista) # Español: Muestra la lista completa de calificaciones / English: Prints the complete list of grades
7     else: # Español: Si aún no se han ingresado 5 calificaciones... / English: If fewer than 5 grades have been entered...
8         return inicio(num) # Español: Llama de nuevo a la función (recursividad) / English: Calls the function again (recursion)
9
10 lista = [] # Español: Crea una lista vacía para guardar las calificaciones / English: Creates an empty list to store the grades
11 global num # Español: Declara la variable 'num' como global (no es realmente necesario) / English: Declares 'num' as global (not really needed)
12 num = 0 # Español: Inicializa 'num' con 0 / English: Initializes 'num' with 0
13
14 if __name__ == '__main__': # Español: Verifica si el archivo se ejecuta directamente / English: Checks if the file is being run directly
15     inicio(num) # Español: Llama a la función 'inicio' para empezar / English: Calls the 'inicio' function to start
16
```

```
Escribe una calificacion
32
Escribe una calificacion
12
Escribe una calificacion
10
Escribe una calificacion
9
Escribe una calificacion
10
[32, 12, 10, 9, 10]
```



## PROGRAMA 2 P2

```

1  from Validaciones import validacion # Español: Importa la clase 'validacion' del módulo 'Validaciones' / English: Imports the 'validacion' class
2  val = validacion() # Español: Crea una instancia de la clase 'validacion' / English: Creates an instance of the 'validacion' class
3
4  class Principal(): # Español: Define una clase llamada 'Principal' / English: Defines a class called 'Principal'
5  def __init__(self): # Español: Constructor de la clase, se ejecuta al crear un objeto / English: Class constructor, runs when an object is created
6      self.lista = [] # Español: Crea una lista vacía para guardar las calificaciones / English: Creates an empty list to store grades
7      self.num = 0 # Español: Inicializa el contador de calificaciones en 0 / English: Initializes the grade counter to 0
8      self.a = "" # Español: Inicializa una variable vacía para almacenar la calificación temporal / English: Initializes an empty variable to store the grade
9
10 def inicio(self): # Español: Define el método 'inicio' que controla el flujo del programa / English: Defines the 'inicio' method that controls the program flow
11     self.a = input('Escribe una calificación \n') # Español: Solicita al usuario que escriba una calificación / English: Asks the user to enter a grade
12     if val.ValidarNumeros(self.a): # Español: Verifica si el valor ingresado es un número válido / English: Checks if the entered value is a valid number
13         self.num += 1 # Español: Aumenta el contador de calificaciones en 1 / English: Increases the grade counter by 1
14         self.lista.append(int(self.a)) # Español: Convierte la calificación a entero y la agrega a la lista / English: Converts the grade to an integer and adds it to the list
15         if self.num >= 5: # Español: Si ya se han ingresado 5 o más calificaciones... / English: If 5 or more grades have been entered...
16             print(self.lista) # Español: Muestra la lista de calificaciones / English: Prints the list of grades
17             print(f'El promedio es: {val.Promedio(self.lista)}') # Español: Muestra el promedio calculado usando la función 'Promedio' / English: Shows the calculated average using the 'Promedio' function
18         else: # Español: Si aún no se han ingresado 5 calificaciones... / English: If fewer than 5 grades have been entered...
19             self.inicio() # Español: Llama de nuevo al método 'inicio' (recursividad) / English: Calls the 'inicio' method again (recursion)
20     else: # Español: Si el valor ingresado no es un número válido... / English: If the entered value is not a valid number...
21         print('No es un número') # Español: Muestra un mensaje de error / English: Displays an error message
22         self.inicio() # Español: Llama nuevamente al método para volver a pedir la calificación / English: Calls the method again to ask for the grade
23
24 if __name__ == '__main__': # Español: Verifica si el archivo se ejecuta directamente (no importado) / English: Checks if the file is being run directly
25     app = Principal() # Español: Crea una instancia (objeto) de la clase 'Principal' / English: Creates an instance (object) of the 'Principal' class
26     app.inicio() # Español: Llama al método 'inicio' para comenzar la ejecución / English: Calls the 'inicio' method to start execution
27

```

```

1  class validacion(): # Español: Define la clase validacion / English: Defines the validation class
2      def __init__(self): # Español: Método constructor / English: Constructor method
3          self.suma = 0 # Español: Inicializa variable suma / English: Initializes sum variable
4          self.promedio = 0.0 # Español: Inicializa variable promedio / English: Initializes average variable
5
6      def ValidarNumeros(self, valor): # Español: Método para validar números / English: Method to validate numbers
7          if valor.isdigit(): # Español: Verifica si el valor son solo dígitos / English: Checks if value contains only digits
8              return True # Español: Retorna verdadero si es número / English: Returns true if it's a number
9          else: # Español: Si no son solo dígitos / English: If not only digits
10             return False # Español: Retorna falso / English: Returns false
11
12     def Promedio(self, lista): # Español: Método para calcular promedio / English: Method to calculate average
13         for i in lista: # Español: Recorre cada elemento de la lista / English: Iterates through each list element
14             self.suma += i # Español: ERROR: debería ser self.suma / English: ERROR: should be self.suma
15         self.promedio = self.suma / len(lista) # Español: Calcula el promedio / English: Calculates the average
16         return self.promedio # Español: Retorna el promedio / English: Returns the average

```

```

Escribe una calificación
20
Escribe una calificación
20
Escribe una calificación
20
Escribe una calificación
20
Escribe una calificación
20
[20, 20, 20, 20, 20]

```

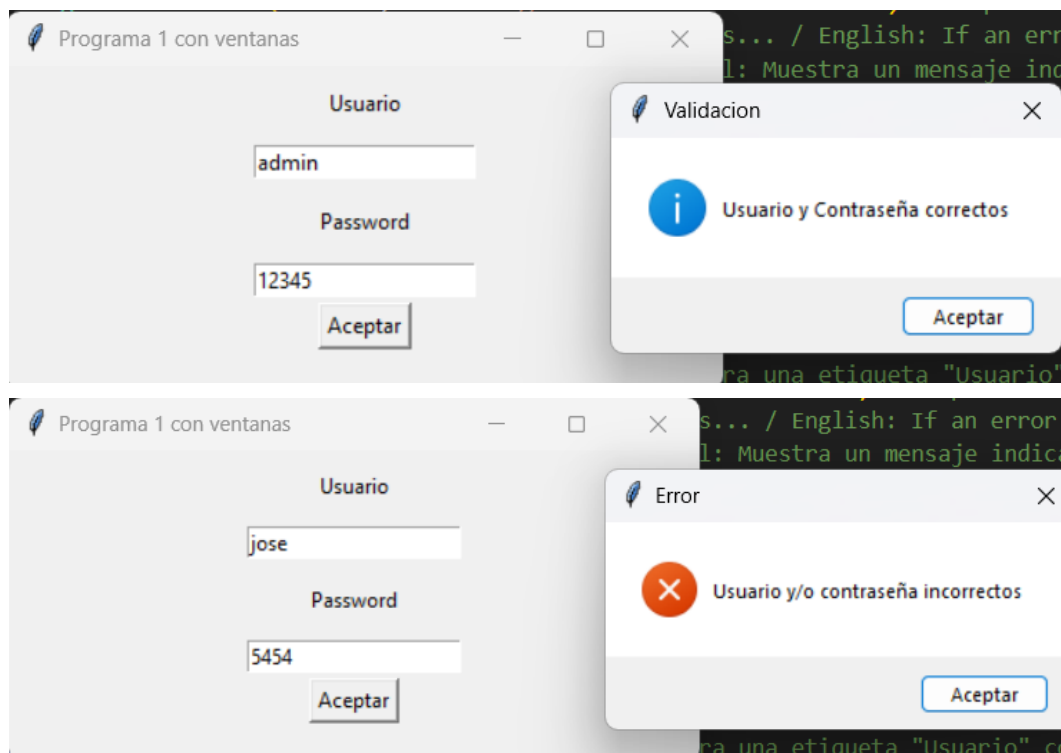


## PROGRAMA 3 P2

```

1 from tkinter import * # Español: Importa todas las clases y funciones del módulo tkinter / English: Imports all classes and functions from the tk
2 from tkinter import messagebox # Español: Importa el módulo messagebox para mostrar cuadros de diálogo / English: Imports the messagebox module
3
4
5 def Ventana(): # Español: Define una función llamada 'Ventana' que crea la interfaz gráfica / English: Defines a function called 'Ventana' that
6
7     def revisar(): # Español: Define una función interna 'revisar' para validar usuario y contraseña / English: Defines an inner function 'revisar'
8         try: # Español: Intenta ejecutar el bloque de código siguiente / English: Tries to execute the following block of code
9             u = str(us.get()) # Español: Obtiene el texto del campo de usuario y lo convierte a cadena / English: Gets the text from the username
10            p = str(pas.get()) # Español: Obtiene el texto del campo de contraseña y lo convierte a cadena / English: Gets the text from the password
11            if u == 'admin' and p == '12345': # Español: Verifica si el usuario y la contraseña son correctos / English: Checks if the username and password are correct
12                messagebox.showinfo('Validacion','Usuario y Contraseña correctos') # Español: Muestra un mensaje de éxito / English: Shows a success message
13            else: # Español: Si los datos no coinciden... / English: If the credentials don't match...
14                messagebox.showerror('Error','Usuario y/o contraseña incorrectos') # Español: Muestra un mensaje de error / English: Displays an error message
15        except: # Español: Si ocurre algún error al obtener los datos... / English: If an error occurs while retrieving data...
16            messagebox.showerror('Error','Introduce datos') # Español: Muestra un mensaje indicando que se deben ingresar datos / English: Displays an error message
17
18    ven = Tk() # Español: Crea la ventana principal de la aplicación / English: Creates the main application window
19    ven.title('Programa 1 con ventanas') # Español: Establece el título de la ventana / English: Sets the window title
20    ven.geometry('400x200') # Español: Define el tamaño de la ventana (ancho x alto) / English: Sets the window size (width x height)
21
22    #label = Label(ven, text='Hola Mundo') # Español: Ejemplo comentado de etiqueta simple / English: Commented example of a simple label
23    #label.pack() # Español: Coloca la etiqueta en la ventana / English: Places the label in the window
24
25    Label(ven, text='Usuario').pack(pady=10) # Español: Crea y muestra una etiqueta "Usuario" con separación vertical / English: Creates and displays a label for the username
26    us = Entry(ven) # Español: Crea el campo de texto para ingresar el usuario / English: Creates the text entry for the username
27    us.pack(pady=3) # Español: Muestra el campo de usuario en la ventana / English: Displays the username field in the window
28    Label(ven, text='Password').pack(pady=10) # Español: Crea y muestra una etiqueta "Password" con separación vertical / English: Creates and displays a label for the password
29    pas = Entry(ven) # Español: Crea el campo de texto para la contraseña / English: Creates the text entry for the password
30    pas.pack(pady=3) # Español: Muestra el campo de contraseña en la ventana / English: Displays the password field in the window
31    boton = Button(ven, text='Aceptar', command=revisar).pack() # Español: Crea un botón que ejecuta 'revisar' al presionarlo / English: Creates an 'Aceptar' button that calls the 'revisar' function
32
33    ven.mainloop() # Español: Mantiene la ventana abierta y en ejecución / English: Keeps the window open and running
34
35 if __name__ == '__main__': # Español: Verifica si el archivo se ejecuta directamente / English: Checks if the file is being run directly
36     Ventana() # Español: Llama a la función 'Ventana' para mostrar la interfaz gráfica / English: Calls the 'Ventana' function to display the GUI
37

```



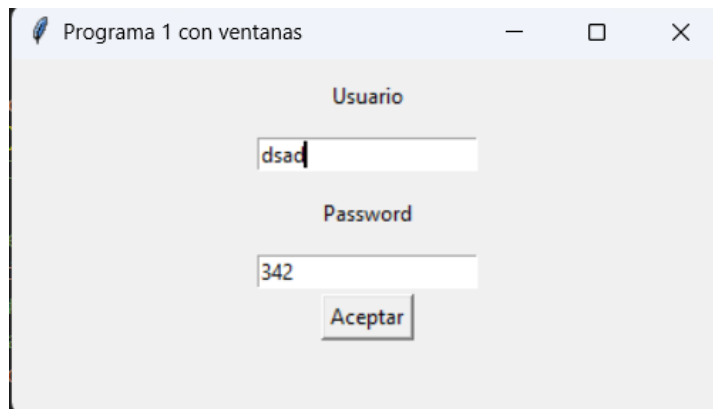


## PROGRAMA 4 P2

```

1 from tkinter import * # Español: Importa todos los módulos de tkinter / English: Imports all tkinter modules
2 from tkinter import messagebox # Español: Importa el módulo de cuadros de mensaje / English: Imports message boxes module
3
4 class Ventana(): # Español: Define la clase Ventana / English: Defines the Window class
5     def __init__(self): # Español: Método constructor / English: Constructor method
6         self.ven = Tk() # Español: Crea la ventana principal / English: Creates main window
7         self.ven.title('Programa 1 con ventanas') # Español: Establece título ventana / English: Sets window title
8         self.ven.geometry('400x200') # Español: Define tamaño ventana / English: Defines window size
9         self.inicio() # Español: Llama método inicio / English: Calls start method
10
11     def inicio(self): # Español: Método para interfaz gráfica / English: Method for GUI interface
12         Label(self.ven, text='Usuario').pack(pady=10) # Español: Crea etiqueta usuario / English: Creates user label
13         self.us = Entry(self.ven) # Español: Crea campo entrada usuario / English: Creates user entry field
14         self.us.pack(pady=3) # Español: Empaqueta campo usuario / English: Packs user field
15         Label(self.ven, text='Password').pack(pady=10) # Español: Crea etiqueta password / English: Creates password label
16         self.pas = Entry(self.ven) # Español: Crea campo entrada password / English: Creates password entry field
17         self.pas.pack(pady=3) # Español: Empaqueta campo password / English: Packs password field
18         boton = Button(self.ven, text='Aceptar', command=self.revisar).pack() # Español: Crea botón aceptar / English: Creates accept button
19         self.ven.mainloop() # Español: Inicia bucle principal / English: Starts main loop
20
21     def revisar(self): # Español: Método validar credenciales / English: Method to validate credentials
22         try: # Español: Bloque try para manejar errores / English: Try block for error handling
23             u = str(self.us.get()) # Español: Obtiene texto usuario / English: Gets user text
24             p = str(self.pas.get()) # Español: Obtiene texto password / English: Gets password text
25             if u == 'admin' and p == '12345': # Español: Verifica credenciales / English: Verifies credentials
26                 messagebox.showinfo('Validacion', 'Usuario y Contraseña correctos') # Español: Mensaje éxito / English: Success message
27             else: # Español: Si credenciales incorrectas / English: If credentials incorrect
28                 messagebox.showerror('Error', 'Usuario y/o contraseña incorrectos') # Español: Mensaje error / English: Error message
29         except: # Español: Si ocurre excepción / English: If exception occurs
30             messagebox.showerror('Error', 'Introduce datos') # Español: Mensaje ingresar datos / English: Enter data message
31
32 if __name__ == '__main__': # Español: Si es archivo principal / English: If main file
33     app = Ventana() # Español: Crea instancia de Ventana / English: Creates Window instance

```



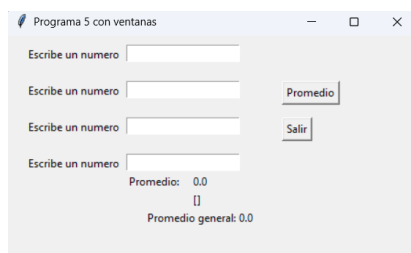


# PROGRAMA 5 P2

```

1  from tkinter import * # Español: Importa todos los módulos de tkinter / English: Imports all tkinter modules
2  from tkinter import messagebox # Español: Importa el módulo de cuadros de mensaje / English: Imports message boxes module
3
4  class Principal(): # Español: Define la clase Principal / English: Defines the Main class
5      def __init__(self): # Español: Método constructor / English: Constructor method
6          self.ven = Tk() # Español: Crea la ventana principal / English: Creates main window
7          self.ven.title('Programa 5 con ventanas') # Español: Establece título ventana / English: Sets window title
8          self.ven.geometry('450x250') # Español: Define tamaño ventana / English: Defines window size
9          self.lista = [] # Español: Crea lista vacía para promedios / English: Creates empty list for averages
10         self.inicio() # Español: Llama método inicio / English: Calls start method
11
12     def sumar(self): # Español: Método para sumar elementos / English: Method to sum elements
13         s = 0 # Español: Inicializa contador en 0 / English: Initializes counter at 0
14         for i in self.lista: # Español: Recorre cada elemento en lista / English: Iterates through each list element
15             s += 1 # Español: Incrementa contador en 1 / English: Increases counter by 1
16         return s # Español: Retorna la suma / English: Returns the sum
17
18     def promediar(self): # Español: Método para calcular promedio / English: Method to calculate average
19         try: # Español: Bloque try para manejar errores / English: Try block for error handling
20             a = float(self.n1.get()) # Español: Convierte primer número a float / English: Converts first number to float
21             b = float(self.n2.get()) # Español: Convierte segundo número a float / English: Converts second number to float
22             c = float(self.n3.get()) # Español: Convierte tercer número a float / English: Converts third number to float
23             d = float(self.n4.get()) # Español: Convierte cuarto número a float / English: Converts fourth number to float
24             pro = (a+b+c+d)/4 # Español: Calcula promedio de 4 números / English: Calculates average of 4 numbers
25             self.l6.config(text=str(pro)) # Español: Actualiza label con promedio / English: Updates label with average
26             self.lista.append(pro) # Español: Agrega promedio a la lista / English: Adds average to the list
27             self.l7.config(text=str(self.lista)) # Español: Muestra lista de promedios / English: Shows averages list
28             self.n1.delete(0,END) # Español: Limpia campo número 1 / English: Clears number 1 field
29             self.n2.delete(0,END) # Español: Limpia campo número 2 / English: Clears number 2 field
30             self.n3.delete(0,END) # Español: Limpia campo número 3 / English: Clears number 3 field
31             self.n4.delete(0,END) # Español: Limpia campo número 4 / English: Clears number 4 field
32             suma = self.sumar() # Español: Obtiene suma de elementos / English: Gets sum of elements
33             p = suma / len(self.lista) # Español: Calcula promedio general / English: Calculates general average
34             self.l8.config(text=f'Promedio General: {str(p)}') # Español: Muestra promedio general / English: Shows general average
35
36         except ValueError: # Español: Captura error de conversión / English: Catches conversion error
37             messagebox.showerror("Error","Algun dato no es numero") # Español: Mensaje error / English: Error message
38             self.n1.delete(0,END) # Español: Limpia campo número 1 / English: Clears number 1 field
39             self.n2.delete(0,END) # Español: Limpia campo número 2 / English: Clears number 2 field
40             self.n3.delete(0,END) # Español: Limpia campo número 3 / English: Clears number 3 field
41             self.n4.delete(0,END) # Español: Limpia campo número 4 / English: Clears number 4 field
42
43     def salir(self): # Español: Método para cerrar ventana / English: Method to close window
44         self.ven.destroy() # Español: Destruye la ventana / English: Destroys the window
45
46     def inicio(self): # Español: Método para interfaz gráfica / English: Method for GUI interface
47         l1 = Label(self.ven, text="Escribe un numero").place(y=10,x=20) # Español: Label número 1 / English: Number 1 label
48         l2 = Label(self.ven, text="Escribe un numero").place(y=50,x=20) # Español: Label número 2 / English: Number 2 label
49         l3 = Label(self.ven, text="Escribe un numero").place(y=90,x=20) # Español: Label número 3 / English: Number 3 label
50         l4 = Label(self.ven, text="Escribe un numero").place(y=130,x=20) # Español: Label número 4 / English: Number 4 label
51         self.n1 = Entry(self.ven) # Español: Campo entrada número 1 / English: Number 1 entry
52         self.n1.place(y=10,x=130) # Español: Posiciona campo 1 / English: Positions field 1
53         self.n2 = Entry(self.ven) # Español: Campo entrada número 2 / English: Number 2 entry
54         self.n2.place(y=50,x=130) # Español: Posiciona campo 2 / English: Positions field 2
55         self.n3 = Entry(self.ven) # Español: Campo entrada número 3 / English: Number 3 entry
56         self.n3.place(y=90,x=130) # Español: Posiciona campo 3 / English: Positions field 3
57         self.n4 = Entry(self.ven) # Español: Campo entrada número 4 / English: Number 4 entry
58         self.n4.place(y=130,x=130) # Español: Posiciona campo 4 / English: Positions field 4
59
60         l5 = Label(self.ven, text="Promedio:").place(y=150,x=130) # Español: Label promedio / English: Average label
61         self.l6 = Label(self.ven, text="0.0") # Español: Label para mostrar promedio / English: Label to show average
62         self.l6.place(y=150,x=200) # Español: Posiciona label promedio / English: Positions average label
63         b1 = Button(self.ven, text="Promedio", command=self.promediar).place(y=50,x=300) # Español: Botón calcular / English: Calculate button
64         b2 = Button(self.ven, text="Salir", command=self.salir).place(y=90,x=300) # Español: Botón salir / English: Exit button
65         self.l7 = Label(self.ven, text="[]") # Español: Label para lista promedios / English: Label for averages list
66         self.l7.place(y=170,x=200) # Español: Posiciona label lista / English: Positions list label
67         self.l8 = Label(self.ven, text="Promedio general: 0.0") # Español: Label promedio general / English: General average label
68         self.l8.place(y=190,x=150) # Español: Posiciona label promedio general / English: Positions general average label
69
70         self.ven.mainloop() # Español: Inicia bucle principal / English: Starts main loop
71
72 if __name__ == '__main__': # Español: Si es archivo principal / English: If main file
73     app = Principal() # Español: Crea instancia de Principal / English: Creates Main instance

```



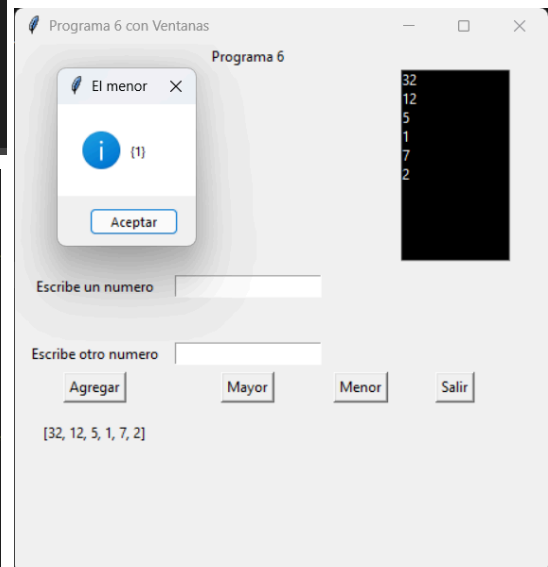


# PROGRAMA 6 P2

```

1 from tkinter import * # Español: Importa todos los módulos de tkinter / English: Imports all tkinter modules
2 from tkinter import messagebox # Español: Importa el módulo de cuadros de mensaje / English: Imports message boxes module
3
4 class Principal(): # Español: Define la clase Principal / English: Defines the Main class
5     def __init__(self): # Español: Método constructor / English: Constructor method
6         self.ven = Tk() # Español: Crea la ventana principal / English: Creates main window
7         self.ven.title("Programa 6 con Ventanas") # Español: Establece título ventana / English: Sets window title
8         self.ven.geometry("450x450") # Español: Define tamaño ventana / English: Defines window size
9         self.a = 0 # Español: Variable para primer número / English: Variable for first number
10        self.b = 0 # Español: Variable para segundo número / English: Variable for second number
11        self.lista = [] # Español: Lista para almacenar números / English: List to store numbers
12        self.aux1 = 0 # Español: Variable auxiliar para mayor / English: Auxiliary variable for maximum
13        self.aux2 = 0 # Español: Variable auxiliar para menor / English: Auxiliary variable for minimum
14        self.cont = 0 # Español: Contador para recursividad / English: Counter for recursion
15
16    def inicio(self): # Español: Método para interfaz gráfica / English: Method for GUI interface
17        l1 = Label(self.ven, text="Programa 6") # Español: Crea título principal / English: Creates main title
18        l1.grid(row=1, column=2) # Español: Posiciona título en grid / English: Positions title in grid
19
20        l2 = Label(self.ven, text="Escribe un número") # Español: Label para primer número / English: Label for first number
21        l2.grid(row=3, column=1, padx=15, pady=10) # Español: Posiciona label en grid / English: Positions label in grid
22
23        Label(self.ven, text="").grid(row=2, column=2) # Español: Label vacío para espacio / English: Empty label for spacing
24
25        self.n1 = Entry(self.ven) # Español: Campo entrada primer número / English: First number entry field
26        self.n1.grid(row=3, column=2) # Español: Posiciona campo en grid / English: Positions field in grid
27
28        l3 = Label(self.ven, text="Escribe otro número") # Español: Label para segundo número / English: Label for second number
29        l3.grid(row=5, column=1, padx=5, pady=5) # Español: Posiciona label en grid / English: Positions label in grid
30
31        Label(self.ven, text="").grid(row=4, column=2) # Español: Label vacío para espacio / English: Empty label for spacing
32
33        self.n2 = Entry(self.ven) # Español: Campo entrada segundo número / English: Second number entry field
34        self.n2.grid(row=5, column=2) # Español: Posiciona campo en grid / English: Positions field in grid
35
36        b1 = Button(self.ven, text="Agregar", command=self.agregar) # Español: Botón agregar números / English: Add numbers button
37        b1.grid(row=6, column=1) # Español: Posiciona botón agregar / English: Positions add button
38        b2 = Button(self.ven, text="Mayor", command=self.mayor) # Español: Botón encontrar mayor / English: Find maximum button
39        b2.grid(row=6, column=2) # Español: Posiciona botón mayor / English: Positions maximum button
40        b3 = Button(self.ven, text="Menor", command=self.menor) # Español: Botón encontrar menor / English: Find minimum button
41        b3.grid(row=6, column=3, padx=10) # Español: Posiciona botón menor / English: Positions minimum button
42
43        b4 = Button(self.ven, text="Salir", command=self.salir) # Español: Botón salir / English: Exit button
44        b4.grid(row=6, column=4, padx=20) # Español: Posiciona botón salir / English: Positions exit button
45        self.listafletores = Label(self.ven, text="") # Español: Label para mostrar lista / English: Label to show list
46        self.listafletores.grid(row=8, column=1, pady=15) # Español: Posiciona label lista / English: Positions list label
47        self.listview = Listbox(self.ven, height=10, # Español: Crea listbox para números / English: Creates Listbox for numbers
48                               width=15, bg="black", # Español: Configura color fondo / English: Configures background color
49                               activestyle="dotbox", fg="white") # Español: Configura estilo y color texto / English: Configures style
50        self.listview.grid(row=2, column=4) # Español: Posiciona Listbox / English: Positions Listbox
51
52        self.ven.mainloop() # Español: Inicia bucle principal / English: Starts main loop
53
54    def mayor(self): # Español: Método para encontrar número mayor / English: Method to find maximum number
55        if len(self.lista) > 0: # Español: Verifica si lista no está vacía / English: Checks if list is not empty
56            #for i in self.lista: # Español: Comentario método alternativo / English: Comment alternative method
57                if self.aux1 < self.lista[self.cont]: # Español: Compara con elemento actual / English: Compares with current element
58                    self.aux1 = self.lista[self.cont] # Español: Actualiza valor mayor / English: Updates maximum value
59                self.cont += 1 # Español: Incrementa contador / English: Increases counter
60            if len(self.lista) - 1 < self.cont: # Español: Verifica fin de lista / English: Checks end of list
61                print(f"el mayor es {self.aux1}") # Español: Imprime mayor en consola / English: Prints maximum in console
62                messagebox.showinfo("El mayor", (self.aux1)) # Español: Muestra mayor en messagebox / English: Shows maximum in messagebox
63                self.cont = 0 # Español: Reinicia contador / English: Resets counter
64            else: # Español: Si no llegó al final / English: If not at the end
65                return self.mayor() # Español: Llama recursivamente / English: Calls recursively
66        else: # Español: Si lista está vacía / English: If list is empty
67            print("lista vacía") # Español: Imprime mensaje consola / English: Prints console message
68            messagebox.showerror("Error", "La lista esta vacía") # Español: Muestra error / English: Shows error
69
70    def menor(self): # Español: Método para encontrar número menor / English: Method to find minimum number
71        if len(self.lista) > 0: # Español: Verifica si lista no está vacía / English: Checks if list is not empty
72            for i in self.lista: # Español: Recorre cada elemento / English: Iterates through each element
73                if self.aux2 > i: # Español: Compara con elemento actual / English: Compares with current element
74                    self.aux2 = i # Español: Actualiza valor menor / English: Updates minimum value
75            print(f"el menor es {self.aux2}") # Español: Imprime menor en consola / English: Prints minimum in console
76            messagebox.showinfo("El menor", (self.aux2)) # Español: Muestra menor en messagebox / English: Shows minimum in messagebox
77        else: # Español: Si lista está vacía / English: If list is empty
78            messagebox.showerror("Error", "La lista esta vacía") # Español: Muestra error / English: Shows error
79
80    def agregar(self): # Español: Método para agregar números / English: Method to add numbers
81        try: # Español: Bloque try para manejar errores / English: Try block for error handling
82            self.a = int(self.n1.get()) # Español: Convierte primer número a entero / English: Converts first number to integer
83            self.lista.append(self.a) # Español: Agrega a lista / English: Adds to list
84            self.listview.insert(self.listview.size()+1, self.a) # Español: Inserta en Listbox / English: Inserts in Listbox
85            self.n1.delete(0, END) # Español: Limpia campo número 1 / English: Clears number 1 field
86            self.b = int(self.n2.get()) # Español: Convierte segundo número a entero / English: Converts second number to integer
87            self.listview.insert(self.listview.size()+1, self.b) # Español: Inserta en listbox / English: Inserts in Listbox
88            self.lista.append(self.b) # Español: Agrega a lista / English: Adds to list
89            self.n2.delete(0, END) # Español: Limpia campo número 2 / English: Clears number 2 field
90            print(self.lista) # Español: Imprime lista en consola / English: Prints list in console
91            self.aux2 = self.lista[0] # Español: Inicializa variable menor / English: Initializes minimum variable
92            self.listafletores.config(text=f"{self.lista}") # Español: Actualiza label con lista / English: Updates label with list
93
94        except ValueError: # Español: Captura error de conversión / English: Catches conversion error
95            messagebox.showerror("Error", "Algun dato no es número") # Español: Muestra error / English: Shows error
96
97    def salir(self): # Español: Método para cerrar ventana / English: Method to close window
98        self.ven.destroy() # Español: Destruye la ventana / English: Destroys the window
99
100
101
102
103 if __name__ == '__main__': # Español: Si es archivo principal / English: If main file
104     app = Principal() # Español: Crea instancia de Principal / English: Creates Main instance
105     app.inicio() # Español: Llama método inicio / English: Calls start method

```





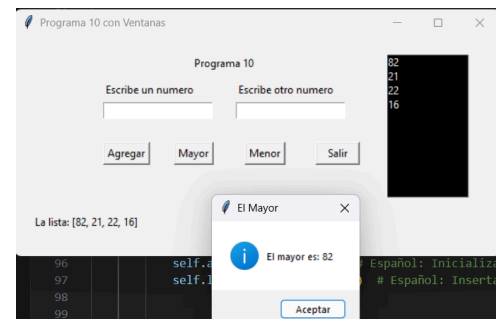


# PROGRAMA 7 P2

```

1 from tkinter import * # Español: Importa todos los módulos de tkinter / English: Imports all tkinter modules
2 from tkinter import messagebox # Español: Importa el módulo de cuadros de mensaje / English: Imports message boxes module
3 import random # Español: Importa librería random / English: Imports random library
4 # Español: importar librería random / English: import random library
5
6 class Principal(): # Español: Define la clase Principal / English: Defines the Main class
7     def __init__(self): # Español: Método constructor / English: Constructor method
8         # Español: Crear ventana principal / English: Create main window
9         self.ven = Tk() # Español: Crea la ventana principal / English: Creates main window
10        self.ven.title("Programa 10 con Ventanas") # Español: Título de la ventana / English: Window title
11        self.ven.geometry("550x250") # Español: Tamaño de la ventana / English: Window size
12        # Español: todo debe ir dentro de aquí / English: everything must go inside this window
13        # Español: pack solo permite un elemento por renglón / English: pack only allows one element per row
14        # Español: Label(ven, text = "Hola mundo").pack() / English: Label(ven, text = "Hello world").pack()
15        self.lista = [] # Español: Crea lista vacía / English: Creates empty list
16        # Español: self.inicio() - llama al método que crea los widgets / English: self.inicio() - calls the method that creates the widgets
17        self.a = 0 # Español: Variable para primer número / English: Variable for first number
18        self.b = 0 # Español: Variable para segundo número / English: Variable for second number
19        self.lista = [] # Español: Lista para almacenar números / English: List to store numbers
20        self.aux1 = 0 # Español: Variable auxiliar para mayor / English: Auxiliary variable for maximum
21        self.aux2 = 0 # Español: Variable auxiliar para menor / English: Auxiliary variable for minimum
22        self.con = 0 # Español: Contador para recursividad / English: Counter for recursion
23
24    def inicio(self): # Español: Método para interfaz gráfica / English: Method for GUI interface
25        # Español: Label(self.ven, text="Programa 9").place(x = 1, y =2) / English: Label(self.ven, text="Program 9").place(x = 1, y =2)
26        l1 = Label(self.ven, text="Programa 10") # Español: Crea título principal / English: Creates main title
27        l1.place(x = 200, y = 20) # Español: Posiciona título / English: Positions title
28        l2 = Label(self.ven, text="Escribe un número") # Español: Label para primer número / English: Label for first number
29        l2.place(x = 100, y = 50) # Español: Posiciona label / English: Positions label
30        self.n1 = Entry(self.ven) # Español: Campo entrada primer número / English: First number entry field
31        self.n1.place(x = 100, y = 75) # Español: Posiciona campo / English: Positions field
32        l3 = Label(self.ven, text="Escribe otro número") # Español: Label para segundo número / English: Label for second number
33        l3.place(x = 250, y = 50) # Español: Posiciona label / English: Positions label
34        self.n2 = Entry(self.ven) # Español: Campo entrada segundo número / English: Second number entry field
35        self.n2.place(x = 250, y = 75) # Español: Posiciona campo / English: Positions field
36        b1 = Button(self.ven, text="Agregar", command=self.agregar) # Español: Botón agregar números / English: Add numbers button
37        b1.place(x=100, y=120) # Español: Posiciona botón agregar / English: Positions add button
38        b2 = Button(self.ven, text="Mayor", command=self.mayor) # Español: Botón encontrar mayor / English: Find maximum button
39        b2.place(x=180, y=120) # Español: Posiciona botón mayor / English: Positions maximum button
40        b3 = Button(self.ven, text="Menor", command=self.menor) # Español: Botón encontrar menor / English: Find minimum button
41        b3.place(x=260, y=120) # Español: Posiciona botón menor / English: Positions minimum button
42
43        b4 = Button(self.ven, text="Salir", command=self.salir) # Español: Botón salir / English: Exit button
44        b4.place(x=300, y=120, width= 50) # Español: Posiciona botón salir / English: Positions exit button
45        self.listaLen = Label(self.ven, text=" ") # Español: Label para mostrar lista / English: Label to show list
46        self.listaLen.place(x = 20, y =200) # Español: Posiciona label lista / English: Positions list label
47        self.listview = Listbox(self.ven, width= 15, bg= "black", activestyle= "dotbox", fg="white") # Español: Crea Listbox para números / English: Creates Listbox for numbers
48        self.listview.place(x=420, y = 20) # Español: Posiciona Listbox / English: Positions Listbox
49        self.ven.mainloop() # Español: Inicia bucle principal / English: Starts main loop
50
51    def salir(self): # Español: Método para cerrar ventana / English: Method to close window
52        self.ven.destroy() # Español: destruimos ventana / English: destroys window
53
54    def mayor(self): # Español: Método para encontrar número mayor / English: Method to find maximum number
55        if len(self.lista) > 0: # Español: Verifica si lista no está vacía / English: Checks if list is not empty
56            if self.aux1 < self.lista[self.con]: # Español: Compara con elemento actual / English: Compares with current element
57                self.aux1 = self.lista[self.con] # Español: Actualiza valor mayor / English: Updates maximum value
58            self.con += 1 # Español: Incrementa contador / English: Increases counter
59            if len(self.lista) == self.con: # Español: Verifica fin de lista / English: Checks end of list
60                messagebox.showinfo("El Mayor", f"El mayor es: {self.aux1}") # Español: Muestra mayor en messagebox / English: Shows maximum in messagebox
61                self.con = 0 # Español: Reinicia contador / English: Resets counter
62            else: # Español: Si no llegó al final / English: If not at the end
63                return self.mayor() # Español: Llama recursivamente / English: Calls recursively
64            # Español: self.con += 1 / English: self.con += 1
65        else: # Español: Si lista está vacía / English: If list is empty
66            messagebox.showerror("Error", "Lista vacía") # Español: Muestra error / English: Shows error
67
68    # Español: def menor(self,n): / English: def menor(self,n):
69    # Español: if self.aux > self.lista[n] / English: if self.aux > self.lista[n]
70
71    def menor(self): # Español: Método para encontrar número menor / English: Method to find minimum number
72        if len(self.lista) > 0: # Español: Verifica si lista no está vacía / English: Checks if list is not empty
73            self.aux2 = self.lista[0] # Español: Inicializa con primer elemento / English: Initializes with first element
74            for i in self.lista: # Español: Recorre cada elemento / English: Iterates through each element
75                if self.aux2 > i: # Español: Compara con elemento actual / English: Compares with current element
76                    self.aux2 = i # Español: Actualiza valor menor / English: Updates minimum value
77            messagebox.showinfo("El Menor", f"El menor es: {self.aux2}") # Español: Muestra menor en messagebox / English: Shows minimum in messagebox
78        else: # Español: Si lista está vacía / English: If list is empty
79            messagebox.showerror("Error", "Lista vacía") # Español: Muestra error / English: Shows error
80
81
82
83    def agregar(self): # Español: Método para agregar números / English: Method to add numbers
84        try: # Español: Bloque try para manejar errores / English: Try block for error handling
85            self.a = random.randint(1,100) # Español: Genera número aleatorio entre 1-100 / English: Generates random number between 1-100
86            self.lista.append(self.a) # Español: Agrega a lista / English: Adds to list
87            self.listview.insert((self.listview.size() + 1), self.a) # Español: Inserta en Listbox / English: Inserts in Listbox
88            # Español: self.n1.delete(0,END) / English: self.n1.delete(0,END)
89            # Español: self.n2.delete(0,END) / English: self.n2.delete(0,END)
90            self.aux1 = int(self.n1.get()) # Español: self.aux1 = int(self.n1.get())
91            self.aux2 = int(self.n2.get()) # Español: self.aux2 = int(self.n2.get())
92            # Español: self.lista.append(self.a) / English: self.lista.append(self.a)
93            # Español: self.listview.insert((self.listview.size() + 1), self.b) / English: self.listview.insert((self.listview.size() + 1), self.b)
94            # Español: self.n2.delete(0,END) / English: self.n2.delete(0,END)
95
96            self.listaLen.config(text=f"La lista: {self.lista}") # Español: Actualiza label con lista / English: Updates label with list
97            self.aux1 = self.lista[0] # Español: Inicializa variable mayor / English: Initializes maximum variable
98            self.listview.insert(self.a) # Español: Inserta número en Listbox / English: Inserts number in Listbox
99
100            # Español: print(self.lista) / English: print(self.lista)
101
102        except ValueError: # Español: Captura error de conversión / English: Catches conversion error
103            messagebox.showerror("Error", "Algun dato no es número") # Español: Muestra error / English: Shows error
104
105
106 if __name__ == "__main__": # Español: Si es archivo principal / English: If main file
107     app = Principal() # Español: Crea instancia de Principal / English: Creates Main instance
108     app.inicio() # Español: Llama método inicio / English: calls start method

```





# REPASO P2

```

6 class Principal(): # Español: Define la clase Principal / English: Defines the Main Class
7     def __init__(self): # Español: Método constructor / English: constructor
8         self.ventana = Tk() # Español: Crea la ventana principal / English: Creates main window
9         self.ventana.geometry("400x200") # Español: Define tamaño ventana / English: Defines window size
10        self.lista = [] # Español: Crea lista vacía / English: Creates empty list
11        self.valid = Validar() # Español: Crea instancia de Validar / English: Creates Validar instance
12
13
14    def inicio(self): # Español: Método para interfaz gráfica / English: Method for GUI interface
15        label(self.ventana, text="Programa de python con Tkinter").place(x=100, y=20) # Español: x = columna y= fila / English: x = column y= row
16        label(self.ventana, text="Escribe un dato").place(x=50, y=50) # Español: Label para entrada dato / English: Label for data entry
17        self.dato = Entry(self.ventana) # Español: Campo entrada de datos / English: Data entry field
18        self.dato.place(x=150, y=50, width=150) # Español: Posiciona campo entrada / English: Positions entry field
19        Button(self.ventana, text="Validar", command=self.validarDatos).place(x=150, y=90, width=150) # Español: Botón validar / English: Validate
20        self.mostrar = Label(self.ventana, text="Ejemplo") # Español: Label para mostrar resultados / English: Label to show results
21        self.mostrar.place(x=20, y=150) # Español: Posiciona label mostrar / English: Positions show label
22        self.ventana.mainloop() # Español: Inicia bucle principal / English: Starts main loop
23
24    def validarDatos(self): # Español: Método para validar datos / English: Method to validate data
25        val = self.dato.get() # Español: Obtiene texto del campo / English: Gets text from field
26        if val != "": # Español: Verifica que no esté vacío / English: Checks it's not empty
27            self.Revisar(val) # Español: Llama método revisar / English: Calls Revisar
28            self.lista.append(val) # Español: Agrega valor a lista / English: Adds value to list
29            self.dato.delete(0,END) # Español: Limpia campo entrada / English: Clears entry field
30            print(self.lista) # Español: Comentario imprimir lista / English: Comment print list
31            self.mostrar.config(text=self.lista) # Español: Actualiza label con lista / English: Updates label with list
32            # respuesta = self.valid.ValidarAscii(val) # Español: Comentario validar ASCII / English: Comment validate ASCII
33            # respuesta = self.valid.ValidarConError(val) # Español: Comentario validar con error / English: Comment validate with error
34            respuesta = self.valid.ValidarConString(val) # Español: Llama validación con string / English: Calls string validation
35            messagebox.showinfo("Validar Datos", f"El dato es {respuesta}") # Español: Muestra resultado / English: Shows result
36        else: # Español: Si campo está vacío / English: If field is empty
37            messagebox.showerror("Error","Caja de texto esta vacía") # Español: Muestra error / English: Shows error
38
39    def Revisar(self, v): # Español: Método para revisar valor / English: Method to check value
40        print(v) # Español: Imprime valor en consola / English: Prints value in console
41
42
43 if __name__ == '__main__': # Español: Si es archivo principal / English: If main file
44     app = Principal() # Español: Crea instancia de Principal / English: Creates Main instance
45     app.inicio() # Español: Llama método inicio / English: Calls start

```

```

class Validar(): # Español: Define la clase Validar / English: Defines the Validate class
    def __init__(self): # Español: Método constructor / English: constructor method
        self.index = 0 # Español: Inicializa variable indice / English: Initializes index variable

    def ValidarAscii(self, valor): # Español: Método para validar usando ASCII / English: Method to validate using ASCII
        con = 0 # Español: Contador para números / English: Counter for numbers
        con2 = 0 # Español: Contador para letras / English: Counter for letters
        for i in valor: # Español: Recorre cada carácter del valor / English: Iterates through each character of value
            if ord(i) >= 48 and ord(i) <= 57: # Español: Verifica si es número (0-9) / English: Checks if it's a number (0-9)
                con += 1 # Español: Incrementa contador números / English: Increases number counter
            if (ord(i) >= 65 and ord(i) <= 90) or (ord(i) >= 97 and ord(i) <= 122): # Español: Verifica si es letra (A-Z o a-z) / English: Checks if it is letter (A-Z or a-z)
                con2 += 1 # Español: Incrementa contador letras / English: Increases letter counter

        if con == len(valor): # Español: Si todos son números / English: If all are numbers
            return "Numeros" # Español: Retorna "Numeros" / English: Returns "Numbers"
        elif con2 == len(valor): # Español: Si todos son letras / English: If all are letters
            return "Letras" # Español: Retorna "Letras" / English: Returns "Letters"
        else: # Español: Si es mezcla / English: If it's a mix
            return "Letras y numeros" # Español: Retorna "Letras y numeros" / English: Returns "Letters and numbers"

    def ValidarConError(self, valor): # Español: Método para validar con manejo de errores / English: Method to validate with error handling
        a = 0 # Español: Variable para manejo de errores / English: Variable for error handling
        b = 0.0 # Español: Variable para Float / English: Variable for float
        try: # Español: Intenta convertir a entero / English: Tries to convert to integer
            a = int(valor) # Español: Conversión a entero / English: Conversion to integer
            return "numeros" # Español: Retorna "numeros" / English: Returns "numbers"
        except ValueError: # Español: Si falla conversión a entero / English: If integer conversion fails
            try: # Español: Intenta convertir a float / English: Tries to convert to float
                b = float(valor) # Español: Conversión a float / English: Conversion to float
                return "Es numero real o con decimales" # Español: Retorna mensaje decimal / English: Returns decimal message
            except ValueError: # Español: Si falla conversión a float / English: If float conversion fails
                return "letras o numeros" # Español: Retorna mensaje texto / English: Returns text message

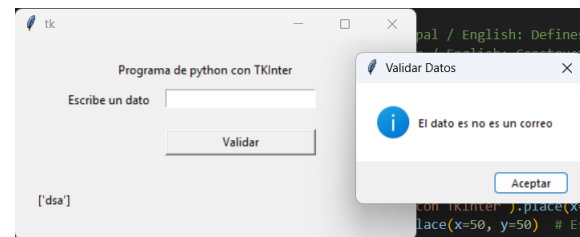
    def ValidarConString(self, valor): # Español: Método para validar correo recursivamente / English: Method to validate email recursively
        print(valor) # Español: Imprime el valor / English: Prints the value
        if self.index < len(valor): # Español: Si no se ha recorrido todo el string / English: If not all string has been traversed
            if valor[self.index] == '@': # Español: Si encuentra el símbolo @ / English: If it finds @
                self.index = 0 # Español: Reinicia indice / English: Resets index
                return "se es un correo" # Español: Retorna que es correo / English: Returns it's an email
            else: # Español: Si no encuentra @ / English: If @ not found
                if self.index < len(valor): # Español: Verifica limite del string / English: Checks string limit

```

```

43         self.index += 1 # Español: Incrementa indice / English: Increases index
44         return self.ValidarConString(valor) # Español: Llama recursivamente / English: Calls recursively
45     else: # Español: Si llegó al final / English: If reached the end
46         self.index = 0 # Español: Reinicia indice / English: Resets index
47         return "no es un correo" # Español: Retorna que no es correo / English: Returns it's not an email
48     else: # Español: Si indice supera longitud / English: If index exceeds length
49         self.index = 0 # Español: Reinicia indice / English: Resets index
50         return "no es un correo" # Español: Retorna que no es correo / English: Returns it's not an email
51
52     # Español: Comentario método alternativo / English: Comment alternative method
53     # @ in valor:
54     # return "si es un correo"
55     # else:
56     #     return "no es in correo"

```







# TAREA P2

```

1 # hacer un programa que los nombre y 3 calificaciones
2 # que se agruparan con las siguientes restricciones
3 # las calificaciones se agruparan en orden descendente,
4 # una vez realizado esto mostrara una pregunta si deseas otra persona
5 # si la respuesta es si, la primer lista se agregara a otra lista que
6 # contenga los 4 datos, para agregar a otra persona.
7
8 from tkinter import * # Español: Importa todos los módulos de tkinter / English: Imports all tkinter modules
9 from tkinter import messagebox # Español: Importa el módulo de cuadros de mensaje / English: Imports message boxes module
10
11 def ventana(): # Español: Define función ventana / English: Defines window function
12     def mostrar(): # Español: Función para mostrar lista final / English: Function to show final list
13         messagebox.showinfo('Lista Final', lista2) # Español: Muestra lista en messagebox / English: Shows list in messagebox
14         print(lista2) # Español: Imprime lista en consola / English: Prints list in console
15
16     def revisar(): # Español: Función para revisar y ordenar calificaciones / English: Function to check and sort grades
17
18     try: # Español: Bloque try para manejar errores / English: Try block for error handling
19         lista2 = [] # Español: Lista temporal para cada estudiante / English: Temporary list for each student
20         n = (nom.get()) # Español: Obtiene nombre del campo / English: Gets name from field
21         c1 = int(cal1.get()) # Español: Obtiene y convierte calificación 1 / English: Gets and converts grade 1
22         c2 = int(cal2.get()) # Español: Obtiene y convierte calificación 2 / English: Gets and converts grade 2
23         c3 = int(cal3.get()) # Español: Obtiene y convierte calificación 3 / English: Gets and converts grade 3
24
25         if (c1 > c2): # Español: Compara calificación 1 y 2 / English: Compares grade 1 and 2
26             if (c1 > c3): # Español: Si calificación 1 es mayor que 3 / English: If grade 1 is greater than 3
27                 lista1.append(n) # Español: Agrega nombre a lista / English: Adds name to list
28                 lista1.append(c1) # Español: Agrega calificación 1 / English: Adds grade 1
29                 if (c2 > c3): # Español: Compara calificación 2 y 3 / English: Compares grade 2 and 3
30                     messagebox.showinfo('Calificaciones', f'es mayor {c1} nes el de en medio {c2} nes el menor {c3}') # Español: Muestra orden
31                     lista1.append(c2) # Español: Agrega calificación 2 / English: Adds grade 2
32                     lista1.append(c3) # Español: Agrega calificación 3 / English: Adds grade 3
33                     lista2.append(lista1) # Español: Agrega lista1 a lista2 / English: Adds lista1 to lista2
34             else: # Español: Si calificación 3 es mayor que 2 / English: If grade 3 is greater than 2
35                 messagebox.showinfo('Calificaciones', f'es mayor {c1} nes el de en medio {c3} nes el menor {c2}') # Español: Muestra orden
36                 lista1.append(c3) # Español: Agrega calificación 3 / English: Adds grade 3
37                 lista1.append(c2) # Español: Agrega calificación 2 / English: Adds grade 2
38                 lista2.append(lista1) # Español: Agrega lista1 a lista2 / English: Adds lista1 to lista2
39             else: # Español: Si calificación 3 es mayor que 1 / English: If grade 3 is greater than 1
40                 messagebox.showinfo('Calificaciones', f'es el de en medio {c1} nes mayor {c3} nes el menor {c2}') # Español: Muestra orden / English: Shows order
41                 lista1.append(n) # Español: Agrega nombre a lista / English: Adds name to list
42                 lista1.append(c3) # Español: Agrega calificación 3 / English: Adds grade 3
43
44                 lista1.append(c1) # Español: Agrega calificación 1 / English: Adds grade 1
45                 lista2.append(lista1) # Español: Agrega lista1 a lista2 / English: Adds lista1 to lista2
46             else: # Español: Si calificación 2 es mayor que 1 / English: If grade 2 is greater than 1
47                 if (c2 > c3): # Español: Si calificación 2 es mayor que 3 / English: If grade 2 is greater than 3
48                     lista1.append(n) # Español: Agrega nombre a lista / English: Adds name to list
49                     lista1.append(c2) # Español: Agrega calificación 2 / English: Adds grade 2
50                     if (c1 > c3): # Español: Compara calificación 1 y 3 / English: Compares grade 1 and 3
51                         messagebox.showinfo('Calificaciones', f'es mayor {c2} nes el de en medio {c1} nes el menor {c3}') # Español: Muestra orden
52                         lista1.append(c1) # Español: Agrega calificación 1 / English: Adds grade 1
53                         lista1.append(c3) # Español: Agrega calificación 3 / English: Adds grade 3
54                         lista2.append(lista1) # Español: Agrega lista1 a lista2 / English: Adds lista1 to lista2
55                     else: # Español: Si calificación 3 es mayor que 1 / English: If grade 3 is greater than 1
56                         messagebox.showinfo('Calificaciones', f'es mayor {c2} nes el de en medio {c1} nes el menor {c1}') # Español: Muestra orden
57                         lista1.append(c3) # Español: Agrega calificación 3 / English: Adds grade 3
58                         lista1.append(c1) # Español: Agrega calificación 1 / English: Adds grade 1
59                         lista2.append(lista1) # Español: Agrega lista1 a lista2 / English: Adds lista1 to lista2
60                     else: # Español: Si calificación 3 es mayor que 2 / English: If grade 3 is greater than 2
61                         messagebox.showinfo('Calificaciones', f'es el de en medio {c2} nes mayor {c3} nes el menor {c1}') # Español: Muestra orden / English: Shows order
62                         lista1.append(n) # Español: Agrega nombre a lista / English: Adds name to list
63                         lista1.append(c3) # Español: Agrega calificación 3 / English: Adds grade 3
64                         lista1.append(c2) # Español: Agrega calificación 2 / English: Adds grade 2
65                         lista1.append(c1) # Español: Agrega calificación 1 / English: Adds grade 1
66                         lista2.append(lista1) # Español: Agrega lista1 a lista2 / English: Adds lista1 to lista2
67                 if len(lista2) == 1: # Español: Si hay al menos un estudiante / English: If there's at least one student
68                     boton2.pack(pady=20) # Español: Muestra botón mostrar lista / English: Shows show list button
69
70     except: # Español: Captura cualquier error / English: Catches any error
71         messagebox.showerror('Error', 'introduce datos') # Español: Muestra error / English: Shows error
72
73     nom.delete(0,END) # Español: Limpia campo nombre / English: Clears name field
74     cal1.delete(0,END) # Español: Limpia campo calificación 1 / English: Clears grade 1 field
75     cal2.delete(0,END) # Español: Limpia campo calificación 2 / English: Clears grade 2 field
76     cal3.delete(0,END) # Español: Limpia campo calificación 3 / English: Clears grade 3 field
77
78
79 ven = Tk() # Español: Crea ventana principal / English: Creates main window
80 ven.title('Programa 1 con ventanas') # Español: Establece titulo ventana / English: Sets window title
81
82 ven.geometry('500x500') # Español: Define tamaño ventana / English: Defines window size
83
84 Label(ven, text='Escribe un nombre').pack(pady=10) # Español: Label para nombre / English: Label for name
85 nom = Entry(ven) # Español: Campo entrada nombre / English: Name entry field
86 nom.pack(pady=3) # Español: Empaqueta campo nombre / English: Packs name field
87 Label(ven, text='Escribe una calificación').pack(pady=10) # Español: Label para calificación 1 / English: Label for grade 1
88 cal1 = Entry(ven) # Español: Campo entrada calificación 1 / English: Grade 1 entry field
89 cal1.pack(pady=3) # Español: Empaqueta campo calificación 1 / English: Packs grade 1 field
90 Label(ven, text='Escribe una calificación').pack(pady=10) # Español: Label para calificación 2 / English: Label for grade 2
91 cal2 = Entry(ven) # Español: Campo entrada calificación 2 / English: Grade 2 entry field
92 cal2.pack(pady=3) # Español: Empaqueta campo calificación 2 / English: Packs grade 2 field
93 Label(ven, text='Escribe una calificación').pack(pady=10) # Español: Label para calificación 3 / English: Label for grade 3
94 cal3 = Entry(ven) # Español: Campo entrada calificación 3 / English: Grade 3 entry field
95 cal3.pack(pady=3) # Español: Empaqueta campo calificación 3 / English: Packs grade 3 field
96
97 boton = Button(ven, text='Confirmar', command=revisar).pack() # Español: Botón confirmar / English: Confirm button
98 boton2 = Button(ven, text='Mostrar lista', command=mostrar).pack() # Español: Botón mostrar lista / English: Show list button
99 ven.mainloop() # Español: Inicia bucle principal / English: Starts main loop
100
101
102
103 lista2 = [] # Español: Lista global para todos los estudiantes / English: Global list for all students
104 if __name__ == '__main__': # Español: Si es archivo principal / English: If main file
105     ventana() # Español: Llama función ventana / English: Calls window function

```

