

EJERCICIO 1

```
1 ##### PROGRAMA PARA LA GESTIÓN DE CALIFICACIONES #####
2
3 # En primer lugar declaramos las funciones
4
5 def menu(): # Función que imprime el menú y devuelve la opción elegida
6     x=input("\nMENÚ DE OPCIONES
7         1- Buscar calificación
8         2- Modificar calificación
9         3- Generar acta
10        4- Alumnado con materia superada
11        5- Alumnado que podrá hacer media
12        6- Salir del programa
13        \n\t\tElige una opción: """)
14     return x
15
16 def presencia(L,E): # Función que, dada una lista y un elemento, devuelve un indicador de presencia del elemento en la lista
17     if E in L:
18         return True
19     else:
20         return False
21
22 def formar_matriz(L1,L2,umbral): # Función que, dadas dos listas y un umbral, construye una matriz
23     # Cada una de las filas de la matriz está formada por los elementos que ocupan la misma posición en las listas de partida,
24     # siempre y cuando el valor numérico en la segunda lista sea superior al umbral dado.
25     superan=[] # matriz en la que se irán almacenando los valores de ambas listas cuando se cumpla la condición
26     for i in range(len(L1)):
27         if L2[i]>umbral:
28             superan.append([L1[i],L2[i]])
29     return superan
30
31 def pedir_dato_num(mensaje,min,max): # Función que devuelve un dato numérico, entre dos dados, correctamente introducido
32     # Esta función no la solicita el enunciado, pero es una tarea que se puede aislar para llamar a la función cuando sea necesaria
33     correcto=False # variable que controla la ejecución del bucle
34     while correcto==False:
35         try: # controla el posible error de ejecución en la conversión del dato
36             dato=float(input(mensaje)) # el primer parámetro es el que permite personalizar el input
37             if dato>=min and dato<=max: # los otros dos parámetros condicionan la validez del dato
38                 correcto=True
39             else:
40                 print("Nota incorrecta")
41         except ValueError:
42             print("Nota incorrecta")
43     return dato
44
45 def imprimir_matriz(M): # Función que, a partir de una matriz bidimensional, construye una cadena para su visualización en pantalla
46     # esta función tampoco la pide el enunciado, pero la usaremos cada vez que se desee imprimir una matriz en "modo bidimensional"
47     cadena=""
48     for i in range(len(M)): # para cada una de las filas de la matriz...
49         for j in M[i]: #... recorremos sus elementos...
50             cadena+=str(j)+"\t" #... y vamos añadiendo a la cadena separados por un tabulador
51         cadena=cadena[0:-1]+\n" # cuando terminamos con una fila, despreciamos el último tabulador y añadimos el cambio de línea
52     # Al finalizar el bucle, la cadena está lista para imprimirse en pantalla
53     print(cadena)
54
55 ##### PROGRAMA PRINCIPAL #####
56
57 nombres=[] # Lista que almacenará los nombres del alumnado
58 notas=[] # Lista que almacenará las notas
59 introducir="S" # variable que controla el bucle de introducción de datos
60 repetir_menu=True # Variable que controla la repetición del menú
61
62 # En primer lugar, se construyen las listas
63 while introducir=="S":
64     nom=input("Nombre: ").title()
65     nota=pedir_dato_num(f"Nota de {nom}: ",0,10) # Llamamos a la función para pedir un dato numérico entre 0 y 10
66     nombres.append(nom)
67     notas.append(nota)
68     introducir=input("Pulsa S para seguir introduciendo datos. Cualquier otra tecla para salir ").upper()
69
70 # Ya cargadas las listas, se presenta el menú, que se repite mientras no se cambia el valor de la variable que lo controla
71 while repetir_menu==True: # bucle que controla la repetición del menú
72     elige=menu()
73     if elige=="1": ##### OPCIÓN 1
74         nombuscado=input("Nombre a buscar: ").title() # pedimos el nombre de la persona cuya nota queremos buscar
75         if presencia(nombres,nombuscado): # averiguamos, llamando a la función, si está en la lista de nombres. Y, si está...
76             notabuscada=notas[nombres.index(nombuscado)] # ... buscamos su posición en la lista para poder localizar la nota que ocupa esa posición
77             print(f"La calificación de {nombuscado} es {notabuscada}") # e imprimimos nombre y nota localizada
78         else: # Si el nombre no está en la lista...
79             print("Esa persona no figura en la lista") # ... imprimimos mensaje de error
80     elif elige=="2": ##### OPCIÓN 2
81         nombuscado=input("Nombre a buscar: ").title() # pedimos el nombre de la persona cuya nota queremos modificar
82         if presencia(nombres,nombuscado): # averiguamos, llamando a la función, si está en la lista de nombres. Y, si está...
83             nota=pedir_dato_num(f"Nueva nota de {nombuscado}: ",0,10) # Llamamos a la función para pedir un dato numérico entre 0 y 10
84             notas[nombres.index(nombuscado)]=nota # y volcamos la nueva nota en la posición correspondiente
85             print(f"La nueva nota de {nombuscado} es {notas[nombres.index(nombuscado)]}") # visualización de comprobación
86         else: # Si el nombre no está en la lista...
87             print("Esa persona no figura en la lista") # ... imprimimos mensaje de error
88     elif elige=="3": ##### OPCIÓN 3
89         acta=formar_matriz(nombres,notas,0) # llamamos a la función para generar el acta. El umbral es 0, puesto que no existe mínimo a superar
90         print("Acta generada")
91         imprimir_matriz(acta) # Llamamos a la función que imprime la matriz volcada en una cadena
92     elif elige=="4": ##### OPCIÓN 4
93         aprueban=formar_matriz(nombres,notas,5) # llamamos a la función para generar el listado. En este caso el umbral es 5
94         print("Lista de personas que superaron la materia")
95         imprimir_matriz(aprueban) # Llamamos a la función que imprime la matriz volcada en una cadena
96     elif elige=="5": ##### OPCIÓN 5
97         hacenMedia=formar_matriz(nombres,notas,4) # llamamos a la función para generar el listado. En este caso el umbral es 4
98         print("Lista de personas que pueden hacer media")
99         imprimir_matriz(hacenMedia) # Llamamos a la función que imprime la matriz volcada en una cadena
100    elif elige=="6": ##### OPCIÓN 6
101        print("Programa finalizado")
102        repetir_menu=False
103    else: ##### Elección errónea
104        print("Elección incorrecta")
105
106
```

EJERCICIO 2

EjMinasFuncionesyListas.py ×

```
def buscar_pos (lista,elemento): # Función que, dada una lista y un elemento, devuelve la posición en la que se encuentra el elemento en la lista
    return lista.index(elemento)

def buscar_elem (lista, pos): # Función que, dada una función y una posición, devuelve el elemento que, en la lista, ocupa la posición especificada.
    for i,j in enumerate(lista):
        if i==pos:
            return j

def VerListas (lista1,lista2): # Función que, dadas dos listas, construye una cadena de visualización.
    cadena_rdo=""
    for i in range(len(lista1)):
        cadena_rdo+=lista1[i]+"\\t"+lista2[i]+"\\n"
        # entre los elementos se concatena un tabulador. Después de ambos elementos, un cambio de línea
    return cadena_rdo

##### PROGRAMA PRINCIPAL #####

# En primer lugar, inicializamos las variables de partida
nombres=["JUAN","ANABEL","JAVIER","SABELA"]
tlfs=["603234567","678987654","698453212","986543210"]
elegir=""
while elegir!="5": # Bucle que controla la repetición del menú. Se repite hasta que la opción elegida sea la quinta.
    elegir=input('1) Añadir una nueva persona a la agenda\\n 2) Buscar por nombre\\n 3) Buscar por teléfono\\n 4) Visualizar la agenda\\n 5) Salir del programa\\n
                Elige una opción =====>> ') # Visualización del menú en el mensaje del input
    # Comienza la estructura de decisión que bifurca el camino de ejecución en función del contenido de la variable elige
    if elegir=="1":
        conseguido=False # variable de control para la petición de nombre
        while conseguido!=True: # bucle que solicita un nombre hasta que sea identificable en la agenda
            nom=input("Nombre de la persona a añadir a la agenda: ").upper() # pedimos el nombre, y lo transformamos a mayúsculas antes de almacenarlo
            if nom in nombres: # si el nombre ya existe en la agenda
                print("Ese nombre está en la agenda. Introdúcelo de nuevo añadiendo algo que lo identifique") # pedimos que lo introduzca de nuevo, con algún distintivo
            else: # si el nombre no está en la agenda...
                conseguido=True # ...lo damos por válido
            tlf=""
            while not tlf.isdigit() or len(tlf)!=9: # bucle que solicita el teléfono hasta su correcta introducción
                tlf=input("Teléfono de contacto de {}: ".format(nom)) # Utilizamos, en el mensaje del input, el nombre de la persona a la que se asocia el teléfono
                if not tlf.isdigit() or len(tlf)!=9: # si el teléfono no es correcto...
                    print("Teléfono introducido incorrectamente") #... imprimimos un mensaje de error
            # Una vez tenemos el nombre y el teléfono, los añadimos a las listas correspondientes
            nombres.append(nom)
            tlfs.append(tlf)
        elif elegir=="2": # Si se elige buscar por nombre...
            nom=input("Nombre de la persona a buscar en la agenda: ") # ...pedimos el nombre a buscar
            if nom in nombres: # Si el nombre dado está en la lista...
                print("El teléfono de {} es: {}".format(nom, buscar_elem(tlfs, buscar_pos(nombres, nom)))) # ... llamamos a las funciones para imprimir el resultado que nos interesa
            else: # Si no está en la lista...
                print("Nombre inexistente en la agenda") #... imprimimos el mensaje correspondiente.
        elif elegir=="3": # De forma similar a lo diseñado en la opción anterior...
            tlf=input("Teléfono a buscar en la agenda: ")
            if tlf in tlfs:
                print("El teléfono {} pertenece a: {}".format(tlf, buscar_elem(nombres, buscar_pos(tlfs, tlf))))
            else:
                print("Teléfono inexistente en la agenda")
        elif elegir=="4": # Si se elige visualizar la agenda...
            print("Agenda actual:\\n", VerListas(nombres, tlfs)) #... se imprime el resultado devuelto por la función que construye la cadena
        elif elegir=="5":
            print("Programa finalizado")
        else: # Si el contenido de la variable elegir es cualquier otra cosa...
            print("Elección incorrecta en el menú") #... la elección sobre el menú ha sido incorrecta.
```