

## ✓ Actividad 7: Gestión de Datos con Diccionarios y Menú Interactivo

Nombre: Alejandro Ramirez Cruz

Matricula: 379551

Fecha: Lunes 27 septiembre de 2025

### ✓ Ejercicio 1: Menú Interactivo para Gestión de Alumnos

#### Descripción:

Desarrollar un programa que permita gestionar datos de alumnos mediante un menú con las siguientes opciones:

1. Leer Diccionario: Generar un diccionario con datos de un alumno, pidiendo al usuario los datos manualmente.
2. Generar Diccionario: Crear un diccionario con datos de un alumno a partir de listas y valores aleatorios.
3. Imprimir Diccionario: Mostrar los datos del alumno en formato de registro.
4. Salir: Finalizar el programa.

#### Explicación del problema:

Se requiere un sistema que permita registrar y consultar la información de alumnos de manera sencilla y segura. Para ello, se implementan funciones de validación que aseguran que los datos introducidos por el usuario sean correctos. Posteriormente, se crean funciones específicas para cada opción del menú. Además, para evitar que el programa falle ante entradas incorrectas o errores inesperados, se utiliza la estructura try-except.

#### Solución:

La solución se implementa dividiendo el programa en bloques de funciones y un menú principal:

- Validación:  
Se crean funciones que verifican que el usuario ingrese datos correctos.  
Para ello se emplea un bucle while True, que mantiene la solicitud activa hasta que la persona ingrese un valor válido.  
También se utilizan métodos como:
  - .strip(): elimina espacios en blanco al inicio y al final de la cadena.
  - .upper(): convierte el texto a mayúsculas, útil para estandarizar respuestas como (H) o (M).
- Gestión de alumnos:  
Se crean funciones que permiten leer los datos directamente del teclado, o generan información aleatoria a partir de listas predefinidas y la biblioteca random.
- Impresión:  
Una función muestra los datos del alumno en formato legible, utilizando f-strings para dar claridad al resultado.
- Menú principal:  
El programa utiliza un bucle while que mantiene activo el menú hasta que el usuario decida salir.  
En cada iteración se muestran las opciones y se emplea .strip() para evitar errores si el usuario agrega espacios, garantizando que la entrada sea procesada correctamente.
- Confirmación de salida:  
Para finalizar el programa, se solicita al usuario que confirme si desea salir ((SI) o (NO)). Se emplea (.strip()) y (.upper()) nuevamente para asegurar que, sin importar cómo escriba la respuesta, sea interpretada de forma correcta.
- Control de errores:  
El uso de try-except en la funciones y en el menú principal evita que el programa se interrumpa en caso de un error inesperado.

```
1 import random # Importar biblioteca para las acciones aleatorias y al azar
```

```
1 # Validacion de entradas
2
3 def validar_entero(mensaje): # Se usa "mensaje" como parametro para reutilizar la funcion y pedir difernete texto
4     while True:
5         try:
6             valor = int(input(mensaje))
7             return valor
8         except ValueError:
9             print("Error: Debe ingresar un número entero, intente de nuevo")
```

```

1 def validar_nombre(mensaje):
2     while True:
3         # .strip() elimina espacios
4         nombre = input(mensaje).strip()
5         # Comprueba que la cadena no este vacia
6         if nombre:
7             return nombre
8         else:
9             print("Error: El nombre no puede estar vacío.")

```

```

1 def validar_edad(mensaje):
2     while True:
3         try:
4             edad = int(input(mensaje)) # Usa el parámetro mensaje
5             if edad >= 18:
6                 return edad
7             else:
8                 print("Error: La edad debe ser mayor o igual a 18 años")
9         except ValueError:
10            print("Error: Debe ingresar un número entero, intente de nuevo")

```

```

1 def validar_sexo(mensaje):
2     while True:
3         # .strip() se usa para limpiar espacio y .upper() para convertir a mayusculas
4         sexo = input(mensaje).strip().upper()
5         if sexo in ['H','M']:
6             return sexo
7         else:
8             print("Error: El sexo debe ser Hombre o Mujer, intente de nuevo")

```

```

1 # Funcion 1: Leer datos del alumno desde el teclado
2
3 def leer_diccionario():
4     print("\n" + "="*100) # Crea un salto de linea y lo une a "=" que imprime una salida mas limpia
5     print("LEER DICCIONARIO: INGRESO MANUAL")
6     print("\n")
7
8     # Solicitar datos con validacion y mensaje de c/u
9     id_alumno = validar_entero("Ingrese ID: ")
10    nombre = validar_nombre("Ingrese nombre(s) 1-2: ")
11    apellido_paterno = validar_nombre("Ingrese apellido paterno: ")
12    apellido_materno = validar_nombre("Ingrese apellido materno: ")
13    edad = validar_edad("Ingrese edad: ")
14    sexo = validar_sexo("Ingrese sexo (H/M): ")
15
16    # Crear lista con los datos
17    lista_datos = [id_alumno, nombre, apellido_paterno, apellido_materno, edad, sexo]
18
19    # Crear diccionario a partir de las lista
20    # Utiliza [#] donde cada clave esta guardada en esa posición de la lista
21    diccionario_alumno = {
22        'ID': lista_datos[0],
23        'Nombre': lista_datos[1],
24        'Apellido paterno': lista_datos[2],
25        'Apellido materno': lista_datos[3],
26        'Edad': lista_datos[4],
27        'Sexo': lista_datos[5],
28    }
29
30    print(";Diccionario creado exitosamente!")
31    return diccionario_alumno

```

```

1 # Función 2: Generar Diccionario con datos automáticamente
2
3 def generar_diccionario():
4     print("\n" + "="*100) # Crea un salto de linea y lo une a "=" que imprime una salida mas limpia
5     print("GENERAR DICCIONARIO: DATOS AUTOMÁTICOS")
6     print("\n")
7
8     # Creación de listas con datos para generar automaticamente
9     nombres = ["Cristiano Ronaldo", "Lionel Andres", "Neymar", "Alisha", "Sofía", "Ericka", "Andrea"]
10    apellidos_pateros = ["Dos Santos", "Messi", "Da Silva", "Martinez", "Gonzales", "Perez", "López"]
11    apellidos_materos = ["Aveiro", "Cuccittini", "Santos", "Sanchez", "Torres", "García", "Ramírez"]
12    sexos = ["H", "M"]
13
14    # Generación de datos aleatorios
15    id_alumno = random.randint(1000,4000)
16    nombre = random.choice(nombres)
17    apellido_paterno = random.choice(apellidos_pateros)
18    apellido_materno = random.choice(apellidos_materos)

```

```

19     edad = random.randint(18,32)
20     sexo = random.choice(sexos)
21
22
23     # Crear lista con los datos
24     lista_datos = [id_alumno, nombre, apellido_paterno, apellido_materno, edad, sexo]
25
26     # Crear diccionario a partir de las lista
27     # Utiliza [#] donde cada clave esta guardada en esa posición de la lista
28     diccionario_alumno = {
29         'ID': lista_datos[0],
30         'Nombre': lista_datos[1],
31         'Apellido paterno': lista_datos[2],
32         'Apellido materno': lista_datos[3],
33         'Edad': lista_datos[4],
34         'Sexo': lista_datos[5],
35     }
36
37     print("¡Diccionario creado exitosamente!")
38     return diccionario_alumno

```

```

1 # Funcion 3: Imprime los datos en formato legible
2
3 def imprimir_diccionario(diccionario_alumno): # Se usa diccionario_alumno para imprimir esta ficha
4     print("\n" + "="*100) # Crea un salto de linea y lo une a "=" que imprime una salida mas limpia
5     print("IMPRIMIR DICCIONARIO")
6     print("\n")
7
8     # Verifica que el diccionario tenga datos disponibles
9     if diccionario_alumno:
10         print("=== DATOS DEL ALUMNO ===")
11         print(f"ID: {diccionario_alumno['ID']}")
12         print(f"Nombre: {diccionario_alumno['Nombre']}")
13         print(f"Apellidos: {diccionario_alumno['Apellido paterno']} {diccionario_alumno['Apellido materno']}")
14         print(f"Edad: {diccionario_alumno['Edad']}")
15         print(f"Sexo: {diccionario_alumno['Sexo']}")
16     else:
17         print("Error: No hay datos disponibles")

```

```

1 # Funcion 4: Salir del programa
2
3 def confirmar_salida():
4
5     while True:
6         # .strip() se usa para limpiar espacio y .upper() para convertir a mayusculas
7         usuario = input("¿Desea salir (SI/NO): ").strip().upper()
8         if usuario == 'SI':
9             return True
10        elif usuario == 'NO':
11            return False
12        else:
13            print("Error: Ingrese 'SI' para salir o 'NO' para continuar")

```

```

1 # Mostrar el menu interactivo
2
3 def mostrar_menu():
4     print("\n" + "="*100) # Crea un salto de linea y lo une a "=" que imprime una salida mas limpia
5     print("=== MENÚ ===")
6     print("1. Leer Diccionario")
7     print("2. Generar Diccionario")
8     print("3. Imprimir Diccionario")
9     print("0. Salir")

```

```

1 # Funcion Principal
2
3 def menu_principal():
4
5     # Variable que guarda la ficha del alumno. None es que esta vacia
6     diccionario_actual = None
7
8     while True:
9         try:
10             mostrar_menu()
11             opcion = input("Seleccione una opcion (0-3): ").strip()
12
13             if opcion == '1':
14                 diccionario_actual = leer_diccionario()
15             elif opcion == '2':
16                 diccionario_actual = generar_diccionario()

```

```

17         elif opcion == '3':
18             imprimir_diccionario(diccionario_actual)
19         elif opcion == '0':
20             if confirmar_salida():
21                 print("Gracias por usar el sistema")
22                 break
23             else:
24                 print("Continuando...")
25         else:
26             print("Error: Opción no valida")
27             continue
28     except KeyboardInterrupt:
29         print("\n\n Programa interrumpido por el usuario.")
30         break

```

```

1 # Ejecuta el programa al ser llamada la función
2 menu_principal()

```

3. Imprimir Diccionario

0. Salir

Seleccione una opcion (0-3): 1

=====

LEER DICCIONARIO: INGRESO MANUAL

Ingrese ID: 379551

Ingrese nombre(s) 1-2: Alejandro

Ingrese apellido paterno: Ramirez

Ingrese apellido materno: Cruz

Ingrese edad: 18

Ingrese sexo (H/M): H

¡Diccionario creado exitosamente!

=====

=== MENÚ ===

1. Leer Diccionario

2. Generar Diccionario

3. Imprimir Diccionario

0. Salir

Seleccione una opcion (0-3): 2

=====

GENERAR DICCIONARIO: DATOS AUTOMÁTICOS

¡Diccionario creado exitosamente!

=====

=== MENÚ ===

1. Leer Diccionario

2. Generar Diccionario

3. Imprimir Diccionario

0. Salir

Seleccione una opcion (0-3): 3

=====

IMPRIMIR DICCIONARIO

=== DATOS DEL ALUMNO ===

ID: 1794

Nombre: Alisha

Apellidos: Perez Aveiro

Edad: 20

Sexo: M

=====

=== MENÚ ===

1. Leer Diccionario

2. Generar Diccionario

3. Imprimir Diccionario

0. Salir

Seleccione una opcion (0-3): 0

¿Desea salir (SI/NO): SI

Gracias por usar el sistema

