

Casos de Estudio

- Mediante un menú de opciones realizar el siguiente programa modular para gestionar el listado de notas de un examen para los estudiantes de una institución educativa:
 - Registrar estudiantes: para cada uno se debe solicitar DNI, nombre y nota. Validar que la nota se encuentre entre 0 y 10. El proceso finaliza cuando se ingresa un DNI igual a cero.
 - Mostrar el listado de estudiantes con sus respectivas notas.
 - Buscar a un estudiante por su DNI y mostrar su nombre y nota.
 - Modificar los datos de un estudiante buscando por DNI (el DNI no se puede modificar).
 - Eliminar a un estudiante por su DNI. Emitir un mensaje de confirmación.
 - Mostrar los estudiantes que obtuvieron nota mayor o igual a una dada y el promedio correspondiente.

Ejercicios

- Analizar, realizar la prueba de escritorio para los valores de entrada solicitados y mostrar la salida de los siguientes códigos:

<pre>s1 = abc aac ede eea cba</pre>	<pre>valores = x y x y x z</pre>
<pre>s1 = input('Ingrese cadena:') s2 = s1.replace(' ','') dic = {} for i in range(len(s2)): letra = s2[i] if letra in dic: dic[letra] += 1 else: dic[letra] = 1 print('Análisis:', '\n', s1, '\n') for key in dic: print(key, '->', dic[key])</pre>	<pre>valores = input('Ingrese cadena:') lst_valores = valores.split(' ') lst_claves = [i for i in range(len(lst_valores))] dic1 = dict(zip(lst_claves, lst_valores)) dic2 = {} for key, value in dic1.items(): if not value in dic2.values(): dic2[key] = value print(dic1) print(dic2)</pre>

- Escribir un programa que implemente una agenda con diccionarios. En la agenda se podrán guardar nombres y números de teléfono. El programa debe mostrar el siguiente menú:
 - Añadir/modificar: solicita un nombre, si el nombre se encuentra en la agenda, debe mostrar el teléfono y, opcionalmente, permitir modificarlo si no es correcto. Si el nombre no se encuentra, debe permitir ingresar el teléfono correspondiente.
 - Buscar: solicitando una cadena de caracteres, y muestra todos los contactos cuyos nombres comienzan por dicha cadena.
 - Borrar: solicita un nombre y si existe nos preguntará si queremos borrarlo de la agenda. Si existe más de uno debe identificarlos por un número secuencial que permita al usuario identificarlo para realizar la operación de borrado.
 - Listar: muestra todos los contactos de la agenda.
- Ingrese un texto y cuente las cantidades de repeticiones para los caracteres que lo componen. Ejemplo de entrada: INTRODUCCION A LA PROGRAMACION y salida:

```
INGRESE TEXTO: INTRODUCCION A LA PROGRAMACION
{'I':3, 'N':3, 'T':1, 'R':3, 'O':4, 'D':1, 'U':1, 'C':3, ' ':3, 'A':4, 'L':1, 'P':1, 'G':1, 'M':1}
```

- Escribir un programa que permita gestionar una base de datos de estudiantes de una carrera de la facultad. Los estudiantes se guardan en un diccionario en el que la clave de cada cliente será su Libreta Universitaria (LU),

los datos del estudiante son nombre, dirección, teléfono, correo, estado), donde el estado tendrá el valor True si se trata de un estudiante activo. El programa debe preguntar al usuario por una opción del siguiente menú:

1. Añadir estudiante, por defecto estado True
 2. Eliminar estudiante,
 3. Mostrar estudiante,
 4. Listar todos los estudiantes,
 5. Listar estudiantes activos o pasivos (permitir la selección)
 6. Cambiar de estado a un estudiante solicitando la LU (True a False o False a True).
5. Utilizando diccionarios diseñar un programa modular que permita gestionar los productos de un comercio, las funcionalidades solicitadas son:
- a. Registrar productos: para cada uno se debe solicitar, código, descripción, precio y stock. Agregar las siguientes validaciones:
 - i. El código no se puede repetir
 - ii. Todos los valores son obligatorios
 - iii. El precio y el stock no pueden ser negativos
 - b. Mostrar el listado de productos
 - c. Mostrar los productos cuyo stock se encuentre en el intervalo [desde, hasta]
 - d. Diseñar un proceso que le sume X al stock de todos los productos cuyo valor actual de stock sea menor al valor Y.
 - e. Eliminar todos los productos cuyo stock sea igual a cero.
 - f. Salir
6. Diseñar un programa que, mediante un menú de opciones, permite realizar lo siguiente:
- a. Cargar dos listas (lista01 y lista02) de contactos de una agenda digital, donde cada contacto tiene número de celular, nombre y correo electrónico.
 - b. Mostrar las listas generadas
 - c. Generar la lista03 que contenga la fusión de las dos primeras, considerar lo siguiente:
 - i. Pueden existir contactos que tengan el mismo número de celular en lista01 y lista02
 - ii. Lista03 puede generarse agregando primero lista01 y luego lista02, pero debe validar que el número de celular no se haya agregado previamente. De ser así debe mostrar un mensaje informando que ya existe el contacto y pedir al usuario que elija cuál de los dos contactos corresponde que se guarde en lista03.
 - d. Salir
7. Realizar un programa para realizar un conjunto de préstamos realizados por una biblioteca. Cada libro en una biblioteca puede ser prestado sólo a un usuario de la biblioteca a la vez. Sin embargo, un usuario puede sacar varios libros. Por lo tanto, la información sobre qué libros se prestan a qué usuarios puede representarse mediante un diccionario, en la que los libros son las claves y los usuarios son los valores. Un ejemplo usando la estructura de datos sería:
- ```
{ "Orgullo y prejuicio" : "Alicia" , "Cumbres Borrascosas" : "Alicia" , "Grandes esperanzas" : "Juan" }
```
- Una operación de búsqueda en la clave "Grandes esperanzas" devolvería "Juan". Si Juan devuelve su libro, eso causaría una operación de eliminación, y si Pat saca un libro, eso causaría una operación de inserción, lo que llevaría a un estado diferente:
- ```
{ "Orgullo y prejuicio" : "Alicia" , "Los hermanos Karamazov" : "Pat" , "Cumbres Borrascosas" : "Alicia" }
```