

## Casos de Estudio

Analice, diseñe y codifique los siguientes enunciados en Python

1. 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:
  - a. 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 el dni es igual a cero.
  - b. Mostrar el listado de estudiantes con sus respectivas notas.
  - c. Buscar a un estudiante por su DNI y mostrar su nombre y nota.
  - d. Modificar los datos de un estudiante buscando por DNI (el DNI no se puede modificar).
  - e. Eliminar a un estudiante buscando por DNI. Emitir un mensaje de confirmación.
  - f. Mostrar el promedio de las notas ingresadas
  - g. Salir

## Ejercicios

1. Considerando las listas del cuadro de la izquierda y en forma manual sin usar la computadora, indique cuál es el resultado y el tipo de las siguientes expresiones. A continuación, verifique sus respuestas en la computadora.

<pre>a = [5, 1, 4, 9, 0] b = list(range(3, 10)) + list(range(20, 23)) c = [[1, 2], [3, 4, 5], [6, 7]] d = ['perro', 'gato', 'jirafa', 'elefante'] e = ['a', a, 2 * a]</pre>	<pre>a[2] b[9] c[1][2] e[0] == e[1] len(c) len(c[0]) len(e) c[-1] c[-1][+1] c[2:] + d[2:] a[3:10] a[3:10:2] d.index('jirafa') e[c[0][1]].count(5)</pre>
---	---

2. Escriba la función `mayores_que(x, lista_valores)` que cuente cuántos valores en la lista valores son mayores que x, por ejemplo `mayores_que(5, [7, 3, 6, 0, 4, 5, 10])` devuelve el valor 3
3. Diseñar un algoritmo que permita realizar lo siguiente:
  - Cargar una lista con N números enteros
  - Mostrar los números ingresados y su posición
  - Mostrar si los elementos de la lista se encuentran ordenados en forma descendente
  - Mostrar los valores que superen el promedio de los valores ingresados
  - Mostrar el mínimo de los valores ingresados y su posición
  - Indicar qué elementos son valores primos
  - El algoritmo debe considerar que si no se cargó la lista previamente, no se pueda realizar alguna de las acciones solicitadas.
4. Diseñar un algoritmo que permita realizar lo siguiente:
  - Cargar una lista con valores de tipo caracter a pedido del operador.
  - Mostrar la lista desde el último valor ingresado hasta el primero.
  - Solicitar un valor al usuario y buscar en la lista devolviendo la posición del primer valor encontrado. En caso que no se encuentre devolver -1
  - Indicar la cantidad de vocales de la lista.

5. Diseñar un algoritmo que permita realizar lo siguiente:
  - Cargar una lista con valores numéricos hasta que el usuario ingrese cero. No debe permitir que se carguen valores duplicados en la lista.
  - Mostrar la lista completa con la cantidad de elementos
  - Agregar un elemento al final de la lista
  - Insertar un elemento preguntando la posición al usuario. Valide que el valor no se encuentre cargado.
  - Eliminar un elemento indicado por el usuario. Si no se encuentra debe informar con un mensaje.
  - Copiar los valores pares a otra lista de nombre listaPares. Mostrar ambas listas.
6. Escribir el código que permita:
  - Cargar una lista con N valores aleatorios (*random.randint(a,b)*), los valores aleatorios deben encontrarse entre 0 y 100. El valor N es el tamaño de la lista ingresado por el usuario.
  - Mostrar el sector de la lista deseada, ingresar el inicio el fin y el paso.
  - Copiar la lista a una denominada listaInversa en donde el orden de los elementos se encuentra en orden inverso a la lista original. Muestre ambas listas.
  - Eliminar de listaInversa los valores duplicados. Mostrar ambas listas
  - Salir
7. Desarrolle una función llamada *desviacion\_estandar(lista\_valores)* de una muestra de una población cuyo parámetro valores sea una lista de números reales. La función debe retornar la desviación estándar de los valores:

$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}$ <ul style="list-style-type: none"> <li>• n es la cantidad de valores de la lista</li> <li>• <math>\bar{x}</math> es el promedio de los valores</li> <li>• <math>x_i</math> son cada uno de los valores</li> </ul>	<p>Hay que realizar las siguientes acciones de forma modular:</p> <ol style="list-style-type: none"> <li>1. calcular el promedio de los valores;</li> <li>2. a cada valor hay que restarle el promedio, y el resultado elevarlo al cuadrado;</li> <li>3. sumar todos los valores obtenidos;</li> <li>4. dividir la suma por la cantidad de valores menos uno; y</li> <li>5. obtener la raíz cuadrada del resultado.</li> </ol> <p>Probar para: lst = [4.0, 1.0, 11.0, 13.0, 2.0, 7.0] desviacion_estandar(lst) #4.88535225615</p>
---	---

8. Escriba un algoritmo para simular un campeonato de tenis.

<p>Primero, debe pedir al usuario que ingrese los nombres de ocho tenistas.</p> <p>A continuación, debe pedir los resultados de los partidos juntando los jugadores de dos en dos. El ganador de cada partido avanza a la ronda siguiente. El algoritmo debe continuar preguntando por los ganadores de partidos hasta que quede un único jugador, que es el campeón del torneo. ¿Puede hacerlo de manera aleatoria?</p>	<p>Jugador 1: Nadal Jugador 2: Melzer Jugador 3: Murray Jugador 4: Soderling Jugador 5: Djokovic Jugador 6: Berdych Jugador 7: Federer Jugador 8: Ferrer</p>	<p>Ronda 1 a.Nadal - b.Melzer: a a.Murray - b.Soderling: b a.Djokovic - b.Berdych: a a.Federer - b.Ferrer: a</p> <p>Ronda 2 a.Nadal - b.Soderling: a a.Djokovic - b.Federer: a</p> <p>Ronda 3 a.Nadal - b.Djokovic: b</p> <p>Campeón: Djokovic</p>
--	--	--