



# EJERCICIOS DE PYTHON II

*Cadenas y Listas*

## Cadenas:

- Ingreso de datos
- Métodos para validar
- Formateo de texto e impresión
- Replicación
- Manipulación
- Segmentación
- Texto multilínea

## Listas:

- Creación y Manipulación. Acceso por subíndice
- Métodos para agregar, buscar, modificar y eliminar elementos
- Recorrer con for, while y for-in
- Concatenación
- Desempaquetado
- Funciones para obtener el largo, sumar elementos, buscar el máximo y el mínimo
- Operadores de pertenencia
- Funciones de ordenamiento

Codo a Codo

2024

- 1) **Informe de Datos Estudiantiles:** La universidad requiere un programa capaz de generar un informe detallado con los datos del estudiante, su nombre de usuario, domicilio, correo electrónico y contraseña. Debes seguir las siguientes pautas:

**Ingreso de datos:**

- El usuario debe ingresar su dni, nombre, apellido y domicilio, atendiendo a las siguientes restricciones:
  - a) DNI: Debe contener solamente números.
  - b) Nombre: No puede tener más de 30 caracteres.
  - c) Apellido: Debe contener únicamente letras y un solo apellido.
  - d) Domicilio: Puede contener cualquier combinación de letras y números en la dirección de calle, pero en el número debe haber al menos un número presente y no tener caracteres alfabéticos.

**Formato del informe:**

- La línea 1 del informe debe contener una fecha alineada a la derecha.
- Las líneas 2 y 4 serán rellenas con 56 caracteres "=", para garantizar el formato adecuado.
- La línea 3 contendrá el string "universidad de python - datos del estudiante", centrado y con el formato "tipo título" que se ve en el modelo.
- A continuación, informar los datos siguiendo estas indicaciones:
  - a) ID: Se genera a partir del DNI y se presenta con una longitud de 11 caracteres, llenando con ceros a la izquierda si es necesario.
  - b) NOMBRE COMPLETO: El apellido debe estar en mayúsculas, se agrega una coma, un espacio y el nombre que debe estar en formato de título (primera letra de cada palabra en mayúscula).
  - c) DOMICILIO: Se presenta en mayúsculas (calle), un espacio y el número.
  - d) USUARIO: El apellido se presenta en minúsculas, se agrega un guion y la primera inicial del nombre en minúsculas, finalmente se acompaña con los primeros tres dígitos del DNI.
  - e) CORREO ELECTRÓNICO: El apellido se presenta en minúsculas, se agrega un punto y el primer nombre\* en minúsculas. Al final se agrega "@unipython.com.ar".  
\* En el caso de nombres compuestos transformar el string en una lista y conservar el primer elemento
  - f) CONTRASEÑA: La primera letra del apellido se presenta en mayúscula, sin dejar espacio se agrega la primera letra del nombre en minúscula, un guion y los últimos 3 dígitos del DNI.
- Dejar una línea en blanco luego de la contraseña y mostrar en 3 líneas utilizando triple comilla simple lo siguiente:

*Recomendamos cambiar su contraseña.*

*La presente información es confidencial.*

*Dpto. de Sistemas.*

**Nota:** Se recomienda seguir el modelo que se acompaña al final de ejercicio y atender a las indicaciones.

Salida por pantalla:

```
31/10/2023
=====
===== Universidad De Python - Datos Del Estudiante =====
=====
ID: 00034521879
NOMBRE COMPLETO: SANCHEZ, Diego Alejandro
DOMICILIO: AV. CORRIENTES 2648
USUARIO: sanchez-d345
CORREO ELECTRÓNICO: sanchez.diego@unipython.com.ar
CONTRASEÑA: Sd-879

Recomendamos cambiar su contraseña.
La presente información es confidencial.
Dpto. de Sistemas.
```

- 2) **Gestión de Materias:** La universidad necesita un programa para gestionar sus cursos, estudiantes y notas. Implementar las siguientes funcionalidades utilizando listas:
- a) Crear una lista llamada Materias que contendrá 3 materias (hardcodeado). Mostrar en pantalla.
  - b) Agregar una materia al final y otra en la segunda posición. Mostrar en pantalla la lista, la posición de la primera materia y la posición de la última.
  - c) Solicitar una materia por teclado. Verificar si existe y en caso de que no exista agregarla al final de la lista, recorrer la lista utilizando un bucle **for**. Mostrar la lista en pantalla y un mensaje de confirmación.
  - d) Solicitar una materia por teclado y modificar su nombre por otro ingresado por teclado. Realizar la búsqueda verificando si la materia existe utilizando un bucle **while**. Mostrar la lista en pantalla y un mensaje de confirmación.
  - e) Solicitar una materia por teclado y eliminarla. Realizar la búsqueda verificando si la materia existe utilizando un bucle **for - in**. Mostrar la lista en pantalla sólo si la materia fue eliminada, junto con un mensaje de confirmación.
  - f) Eliminar la última materia de la lista y la primera materia de la lista. En ambos casos verificar que la lista no esté vacía.

**Nota:** Las operaciones de este ejercicio están relacionadas con lo que se conoce como un sistema CRUD (Create, Read, Update, Delete) en el contexto de la gestión de datos:

➤ **Create:**

- b) Agregar una materia al final y otra en la segunda posición.
- c) Solicitar una materia por teclado. Verificar si existe y agregarla al final si no existe.

➤ **Read (Leer):**

- a) Mostrar la lista de materias.

**Update (Actualizar):**

- d) Solicitar una materia por teclado y modificar su nombre por otro ingresado por teclado.

➤ **Delete (Eliminar):**

- e) Solicitar una materia por teclado y eliminarla.
- f) Eliminar la última materia de la lista y la primera materia de la lista.

Todas estas operaciones forman parte de un conjunto completo de acciones que permiten crear, leer, actualizar y eliminar datos en una lista de materias.

Posible salida por pantalla:

```
['Python I', 'Bases de datos I', 'Filosofía']
Lista: ['Python I', 'Python II', 'Bases de datos I',
'Filosofía', 'Pseudocódigo']
Primera materia: Python I
Última materia: Pseudocódigo
['Python I', 'Python II', 'Bases de datos I',
'Filosofía', 'Pseudocódigo']
Ingrese el nombre de la materia: Python III
Materia Python III agregada.
Lista actualizada: ['Python I', 'Python II', 'Bases de
datos I', 'Filosofía', 'Pseudocódigo', 'Python III']
Ingrese el nombre de la materia a modificar: Python I
Ingrese el nuevo nombre para la materia: Introd. a
Python
Materia Python I modificada a Introd. a Python.
Ingrese el nombre de la materia a eliminar: Pseudocódigo
Materia Pseudocódigo eliminada.
Lista actualizada de materias: ['Introd. a Python',
'Python II', 'Bases de datos I', 'Filosofía', 'Python
III']Se eliminó la última materia: Python III
Se eliminó la primera materia: Introd. a Python
```

### 3) Estadísticas sobre edades de alumnos:

- Crear dos listas que contengan edades de un grupo de estudiantes, llamarlas **lista1** y **lista2**. Concatenarlas en una lista llamada **edades**. Mostrar las tres listas.
- Determinar la edad más alta y mostrarla en pantalla.
- Determinar la edad más baja y mostrarla en pantalla.
- Obtener la posición de la edad más baja.
- Obtener el promedio de las edades.
- Contar cuántos estudiantes tienen alguna edad que **no** se encuentre en la lista.
- Contar cuántos estudiantes tienen alguna edad que se encuentre en la lista.
- Invertir el orden de las edades y mostrarlas en pantalla.
- Ordenar las edades de menor a mayor y mostrarlas en pantalla.
- Ordenar las edades de mayor a menor y mostrarlas en pantalla.
- Borrar la lista de edades y mostrarla en pantalla.

#### Extra: Desempaquetado y pertenencia

- Crear una lista llamada turnos que contenga los valores Mañana, Tarde y Vespertino. Verificar la pertenencia de "Noche" en la lista y "Mañana" en la lista. Imprimir los resultados.
- Desempaquetar la lista y mostrar los resultados individuales.

Salida por pantalla:

```
Primera lista: [26, 38, 47, 30, 50, 19, 50, 48]
Segunda lista: [35, 40, 45, 55, 21]
Listas unidas: [26, 38, 47, 30, 50, 19, 50, 48, 35, 40,
45, 55, 21]
Edad más alta: 55
Edad más baja: 19
Posición de la edad más baja: 5
```

```
Promedio de edades: 38.77
Cantidad de 18 años: 0
Cantidad de 50 años: 2
Edades invertidas: [21, 55, 45, 40, 35, 48, 50, 19, 50,
30, 47, 38, 26]
Edades ordenadas: [19, 21, 26, 30, 35, 38, 40, 45, 47,
48, 50, 50, 55]
Edades ordenadas al revés: [55, 50, 50, 48, 47, 45, 40,
38, 35, 30, 26, 21, 19]
Lista vacía: []
Lista de turnos: ['Mañana', 'Tarde', 'Vespertino']
Noche está en la lista? False
Mañana está en la lista? True
Turno 1: Mañana
Turno 2: Tarde
Turno 3: Vespertino
```