

# Práctica 5. Tipos de datos estructurados en Python

## 1. Objetivos

- Aprender a hacer programas que utilicen los diferentes objetos de datos estructurados que ofrece Python: cadenas, listas y diccionarios.
- Implementar las operaciones administrativas básicas y utilizar algunos métodos asociados al manejo de diccionarios en Python e implementar las operaciones requeridas para administrar las listas.
- Consolidar el desarrollo de programas en Python siguiendo las directrices de la programación modular utilizando funciones.
- Practicar la interacción de los programas con archivos de texto para leer y escribir datos.
- Aprender algoritmos que procesan archivos JSON y CSV como formatos estándar en transmisión y almacenamiento de datos.

**Nota:** A pesar de que es parte de su labor decidir las estructuras de datos a utilizar para construir sus programas, en este enunciado se le indicará explícitamente cuáles utilizar en algunos casos puntuales.

## 2. Introducción

En esta práctica se desarrollará un programa que permita hacer la lectura de diferentes variables con el fin de realizar el monitoreo del clima. La administración departamental, consciente de los problemas que el cambio climático y la contaminación del aire puede causar a sus habitantes, decidió iniciar un proyecto para instalar estaciones de monitoreo de variables ambientales y de calidad del aire en las diferentes subregiones del departamento, para recolectar datos que sirvan de insumo para diseñar políticas públicas que permitan mitigar los efectos de estos fenómenos en la población.

## 3. Requerimientos del programa

### 3.1. Requisitos funcionales

a) **Usuarios:** Para almacenar los datos recolectados en estas estaciones, se debe diseñar un sistema de información, que podrá ser usado por usuarios registrados y visitantes.

- **Usuario visitante:** Puede visualizar estadísticas básicas acerca de los datos registrados en el sistema.

- **Usuario registrado:** Debe identificarse con documento y contraseña para acceder al sistema. Pueden realizar las siguientes acciones, según su rol o perfil:
  - **Administrador:** puede crear, editar y eliminar usuarios y estaciones del sistema.
  - **Operador:** Puede ingresar al sistema la información de las estaciones de monitoreo y también puede visualizarla.

**b) Base de datos:** El sistema tendrá una base de datos en la que se almacenará la información de los usuarios registrados, las estaciones existentes en el sistema, las variables monitoreadas y el registro de las medidas ingresadas al sistema. Esta información se guardará de manera persistente en archivos con formato JSON y CSV, pero debe ser leída por el programa y cargada en memoria, usando estructuras de datos, para su manipulación durante la ejecución del programa, y su posterior actualización en el archivo de texto. La estructura (formato) de los archivos JSON y CSV es la que se muestra a continuación:

```
mediciones.json
{
  "codigo_estacion_1" :
  {
    "variable_1" : {
      "YYYY-MM-DD" : [value_1, value_2, ...],
      "YYYY-MM-DD" : [value_1, value_2, ...],
      ...
    },
    ...
    "variable_n" : {
      "YYYY-MM-DD" : [value_1, value_2, ...],
      "YYYY-MM-DD" : [value_1, value_2, ...],
      ...
    },
    ...
  },
  ...
  "codigo_estacion_n" : {...}
}
```

```
estaciones_config.csv
Código de Estación, Nombre
CODIGO_ESTACION1, nombre
...
CODIGO_ESTACION_n, nombre
```

users.csv

```
identificacion, nombre, clave, rol  
id_1, nombre_1, password_1, rol  
...  
id_n, nombre_n, password_n, rol
```

variables\_config.json

```
{  
  "variables":  
    {  
      "var_1" :  
        {  
          "minimo" : value,  
          "maximo" : value,  
          "unidades" : value  
        },  
      ...  
      "var_n" :  
        {  
          ...  
        }  
    }  
}
```

**Nota:** Las estaciones llevan a cabo un número constante de mediciones diarias, aunque estas pueden ocurrir en diferentes días y, en ocasiones, puede haber días sin mediciones.

### c) Menús y operaciones del sistema

Menú inicial:

- Usuario registrado
- Usuario visitante
- Salir del sistema

Si se elige la opción de **Usuario registrado**, se le pedirá al usuario que se autentique con documento y contraseña, para validar que sea un usuario registrado en el sistema. Una vez se autentique el usuario, se le mostrará el menú correspondiente a su rol o perfil en el sistema:

### Usuario Operador:

- Seleccionar la estación:
  - Mostrar medidas: Mostrar en una tabla los valores registrados para esa estación.
  - Ingresar medidas.
    - El sistema tomará automáticamente la fecha y hora actual para la medida.
    - Ingresar los valores de cada una de las variables del sistema.
      - Se debe validar que el valor ingresado esté en el rango válido de la variable correspondiente (mínimo  $\leq$  valor  $\leq$  máximo).
      - En caso de que el valor no esté disponible (falta o fallo del sensor en un momento dado), se debe ingresar 'ND' y el sistema debe registrarlo en la base de datos con el valor numérico -999, que será el indicativo de valor no disponible o no válido.

### Usuario Administrador:

- Volver al menú inicial.
- Gestionar estaciones:
  - Crear estación
    - Ingresar el nombre de la estación. El código de la estación se debe generar automáticamente (código numérico incremental).
  - Editar estación:
    - Elegir la estación del listado.
    - Mostrar el nombre actual de la estación e ingresar el nuevo. El código de la estación no se modifica.
  - Eliminar estación:
    - Elegir la estación del listado. Solo puede eliminarse una estación cuando no hay ningún registro de valores asociado a ella.
- Gestionar usuario:
  - Crear usuario:
    - Ingresar documento, que debe ser diferente a todos los anteriormente registrados (no puede haber 2 documentos iguales). Debe validarse que contenga 10 caracteres numéricos (solamente puede tener números).
    - Ingresar nombre completo. El nombre solo puede contener letras y espacios.
    - Ingresar la contraseña, que debe contener mínimo 4 caracteres.
    - Confirmar la contraseña, validar que sea igual a la anteriormente escrita.
    - Seleccionar el rol para el usuario (Administrador u Operador).
  - Editar usuario:
    - Elegir el usuario del listado.

- Mostrar el nombre actual e ingresar el nuevo nombre.
- Mostrar la contraseña actual e ingresar la nueva contraseña.
- Confirmar la nueva contraseña, validar que sea igual a la anteriormente escrita.
- Mostrar el rol actual y seleccionar el nuevo rol para el usuario.
- Eliminar usuario:
  - Elegir el usuario del listado.
  - Confirmar que se desea eliminar el usuario. No puede eliminarse el usuario actual del sistema.
- Depuración de registros inconsistentes en la base de datos:

Suponga que el archivo de datos se ha duplicado accidentalmente y se tienen dos versiones muy similares del mismo. Por una parte, la versión original que usted ha usado en su programa hasta este punto, y la segunda versión llamada igual, pero finaliza en “v2”. Ambas versiones tienen exactamente los mismos datos según el mismo formato (no necesita validar esto), pero presentan diferencias en los registros de las mediciones (pudiendo sobrar o faltar algún registro). Es necesario entonces, leer el archivo duplicado (además del que ya se había leído, el original), y comparar los registros, generando por pantalla dos reportes diferentes:

- Reportar los registros de las medidas comunes en ambos archivos (que aparecen en ambos a la vez y sin repetirlos en la pantalla).
- Reportar los registros de las medidas que aparecen en cualquiera de los dos archivos. Si un registro aparece en ambos archivos, en el reporte sólo debe aparecer una sola vez.

**Nota:** En las funcionalidades de eliminar registro, las funciones deben ser implementadas por cada grupo, no se permite el uso de los métodos de Python.

Si se elige la opción de **Usuario visitante**, se despliega el siguiente menú:

- Visualizar estadísticas
  - Elegir el periodo de tiempo a evaluar:
    - Últimos 7 días
    - Últimos 30 días
    - Elegir fechas manualmente
  - Elige las variables a analizar del listado. Puede elegir una, varias o todas las variables.
  - Elegir modo de visualización: los resultados se pueden (1) visualizar por pantalla o (2) almacenarse en el archivo de texto plano `Estadisticas.txt`. El formato de `Estadisticas.txt` es libre, a su elección.

Las estadísticas que se mostrarán serán: el valor máximo, mínimo y promedio de las medidas tomadas en las estaciones de los municipios seleccionados y en el lapso de tiempo seleccionado. Para el máximo y mínimo, se debe mostrar también el nombre de la estación en que se registró la medida, así como la fecha. En caso de que se elijan más de una variable, el sistema debe mostrar (sea por pantalla o por archivo) los resultados para cada variable, una por una, para permitir que el usuario las visualice fácilmente.

#### d) Validaciones necesarias:

El sistema debe realizar las siguientes validaciones para su correcto funcionamiento:

- Fecha: debe tener formato yyyy-mm-dd, validar que los días correspondan con el mes correspondiente, incluso para años bisiestos.
- Nombre: Solo puede contener letras y espacios.
- Documento: Debe tener 10 caracteres, todos numéricos.
- Contraseña: Debe tener mínimo 4 caracteres.
- Valor de la medida: Debe ser un número y estar dentro del rango definido para la variable respectiva. La única excepción es el valor que indica valor No Disponible (-999 ó 'ND', según se elija).

**Pista:** use las funciones `int()` ó `float()` para la conversión de datos.

### 3.2. Requisitos no funcionales

El programa desarrollado debe cumplir con lo siguiente:

- Utilizar funciones en el desarrollo del programa a criterio del programador con sus respectivas especificaciones a manera de docstrings.
- Debe validar las entradas por teclado a nivel de tipo y de rango (cuando así corresponda).
- Crear al menos dos módulos (archivos fuente), uno para el programa principal que implementará los menús para la interacción con el usuario, y otro para las funciones que se encargarán de la administración de la información de las bases de datos.
- Todos los métodos restringidos según se indicó en las clases de teoría, no pueden usarse en el desarrollo de la solución.
- Utilizar strings, listas y diccionarios, según sea conveniente. No se admiten otros tipos de datos o métodos avanzados de Python que no se hayan visto en clase.

## 4. Etapas de desarrollo sugeridas

- Al finalizar la primera sesión, se debe entregar un diagrama de flujo general (sin muchos detalles) del funcionamiento general del sistema y un diagrama libre que indique cómo se van a organizar los datos en el programa: tipos de variables, jerarquías, contenidos, etc. Estos son los insumos principales para poder iniciar con el desarrollo del código y debe entregarse una copia al docente, que hará parte de la evaluación de la práctica.
- Para el inicio de la segunda sesión, se deben tener lista definición de las funciones que se van a utilizar en el sistema, para lo cual es muy útil el diagrama de flujo realizado anteriormente, y la implementación que aquellas que realicen las validaciones requeridas en el sistema.
- Al finalizar la segunda sesión se deben tener listas las funciones necesarias para cargar en las variables los datos de los archivos JSON y CSV y para el proceso inverso, es decir, guardar en los archivos todo lo que esté en las variables, respetando el formato definido. También deben quedar listas las funciones para listar, crear, editar y eliminar usuarios en el sistema.

**Nota:** Para cargar los datos de los archivos puede hacer uso de la librería `json`.

Mientras que para el proceso de guardar, debe implementarse manualmente.

- Al inicio de la tercera sesión se debe tener listas las funciones para listar, crear, editar y eliminar estaciones en el sistema, así como las funciones para ingresar y listar medidas de las estaciones y las que permitan encontrar los valores máximo, mínimo y promedio de una variable en un periodo de tiempo.
- Al final de la tercera sesión se debe tener lista la función que permita validar el acceso de los usuarios registrados y los demás menús del sistema.
- En la fecha límite establecida por el docente se debe entregar el programa completamente funcional, que cumpla con todos los requerimientos funcionales y no funcionales.

## 5. Evaluación

La evaluación se basará en el código entregado, una sustentación oral sobre el mismo y los temas cubiertos en la práctica. Además, se tendrá en cuenta una bonificación para quien haga la parte opcional.

**Sugerencia:** Se proveen dos archivos adicionales para esta práctica. En `utilidades.py` encontrará docstrings con algunas de las funciones que se recomienda implementar. Además, encontrará terminadas algunas implementaciones que el estudiante puede reusar de manera responsable. Es decir, puede reusarlas al tiempo que comprenda el funcionamiento general de

las mismas. En `ejemplo_utilidad.py` se muestra como reusarlas. El uso de estos archivos es totalmente opcional.