



Práctica 6: Tipos de datos estructurados en Python

1. Objetivos

- Aprender a hacer programas que utilicen los diferentes objetos de datos estructurados que ofrece Python: cadenas, tuplas, listas y diccionarios.
- Conocer y utilizar los métodos (funciones) asociados a los datos estructurados en Python.
- 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.

2. Introducción

Esta práctica se desarrollará un programa asociado con 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.

2.1. Requerimientos del programa

Requisitos funcionales

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.

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, los municipios y el registro de las medidas ingresadas al sistema. Esta información se guardará de manera persistente en un archivo de texto, pero debe ser leída por el programa y cargada memoria, usando estructuras

de datos, para su manipulación durante la ejecución del programa, y su posterior actualización en el archivo de texto. La estructura del archivo de texto es la que se muestra a continuación:

```
<DOCUMENTO1;NOMBRE_COMPLETO;CONTRASEÑA;ROL>
...
<DOCUMENTON;NOMBRE_COMPLETO;CONTRASEÑA;ROL>
:MUNICIPIO1,MUNICIPIO2,MUNICIPIO3,...,MUNICIPIO_N
CODIGO_ESTACION,NOMBRE_ESTACION,MUNICIPIO
VARIABLE1 [MINIMO:MAXIMO,UNIDAD];...;VARIABLE_N [MINIMO:MAXIMO,UNIDAD]
YYYY-MM-DD HH:MM:SS;CODIGO_ESTACION;{VALOR_VARIABLE1,...,VALOR_VARIABLE_N}
...
YYYY-MM-DD HH:MM:SS;CODIGO_ESTACION;{VALOR_VARIABLE1,...,VALOR_VARIABLE_N}
```

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 el municipio en que está ubicada la estación o volver al menú inicial.
- Seleccionar la estación del listado de estaciones correspondientes al municipio seleccionado o volver a la opción de elegir el municipio.
 - Listar 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 (máximo < valor < mínimo).
 - 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).
 - Seleccionar el municipio de la lista.
 - Editar estación:
 - Elegir la estación de la lista.

- Mostrar el nombre actual de la estación e ingresar el nuevo. El código de la estación no se modifica.
- Mostrar el municipio actual y seleccionar el nuevo municipio de la lista.
- Eliminar estación:
 - Elegir la estación de la lista. 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 numérico (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 de la lista.
 - 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 de la lista.
 - Confirmar que se desea eliminar el usuario. No puede eliminarse el usuario actual del sistema.

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 los municipios para el análisis. Puede elegir uno, varios o todos los municipios.

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 y hora registrada. En caso de que se elijan más de una variable, el sistema debe mostrar los resultados para cada variable, una por una, para permitir que el usuario las visualice fácilmente.

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 `isalpha()`, `isnumeric()` y `replace()` de los string y la conversión de datos usando `int()` ó `float()`.

Requisitos no funcionales

El programa desarrollado debe cumplir con las siguientes:

- Utilizar funciones en el desarrollo del programa a criterio del programador con sus respectivas especificaciones a manera de docstrings.
- Crear 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.
- Utilizar strings, tuplas, listas y diccionarios.
- Realizar pruebas de caja negra y caja blanca, como mínimo, a las funciones de validación de datos requeridas.

Opcional

Agregar a las estadísticas un ranking de las 5 horas del día con mayor concentración de cada una de las variables seleccionadas.

2.2. 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 de 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 del archivo de texto 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.
- 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, junto con las pruebas de caja blanca y caja negra de las funciones para hallar los valores máximo, mínimo y promedio de una variable, en un intervalo de tiempo dado y para los municipios seleccionados.

3. Evaluación

La evaluación se basará en el código entregado, incluyendo las pruebas de caja blanca y caja negra realizadas, y 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.