



# Práctica 7: 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. Tareas a realizar

Esta práctica consiste en el desarrollo de un programa para consultar estadísticas de operación del sistema **metro**. El programa está pensado para ser utilizado por la administración del metro con el objetivo de conocer detalles sobre el flujo de pasajeros en un día. El programa debe ser desarrollado de manera modular mediante el uso de funciones de acuerdo al mejor criterio posible.

### 1.1. Resumen de la aplicación

El programa a desarrollar tiene como función principal, extraer información cruda de un archivo que contiene el reporte de todos los viajes realizados en un día de operación del metro. El programa debe ofrecer, mediante un menú jerárquico, la consulta de diferentes estadísticas sobre el flujo de pasajeros en el sistema.

### 1.2. Interfaz de usuario

El programa debe interactuar con el usuario mediante dos tipos de menú. El primer menú que debe mostrarse le permitirá al usuario seleccionar de entre dos tipos de estadísticas a consultar: generales del sistema o específicas por estación. Luego, dependiendo de la opción elegida, se debe mostrar un nuevo menú con las opciones disponibles en cada caso. Cuando la opción seleccionada en el menú principal sea la de estadísticas por estación, el programa deberá preguntar al usuario por el nombre o código de la estación de interés. Cada menú deberá ofrecer además la opción de salir. A continuación se muestra cómo debe ser cada uno de los menús:

#### Menú principal

Sistema de consulta de estadísticas de operación del metro

1. Estadísticas generales del sistema

2. Estadísticas por estación

Ingresa el número con la opción deseada (o x para salir):

### **Menú para la opción 1**

1. Cantidad de personas que usaron el metro
  2. Cantidad de viajes que hubo
  3. Hora pico de ingreso de usuarios
  4. Ingresos totales por concepto de tiquetes
  5. Estaciones en las que más personas ingresan y salen
  6. Distancia y tiempo promedio de viaje de los usuarios
- Ingrese el número con la opción deseada (o x para salir):

### **Pregunta previa para el menú 2**

Ingrese el nombre o el código de la estación de interés:

### **Menú para la opción 2**

1. Cantidad total de viajes
  2. Horas pico de ingreso y de salida de usuarios
  3. Estaciones de origen y destino más comunes
- Ingrese el número con la opción deseada (o x para salir):

## **1.3. Formato de los archivos de datos**

Al iniciar, el programa debe cargar en variables toda la información que se encuentra en dos archivos que se describen a continuación.

### **stations.info**

Este archivo contiene los nombres de las estaciones del metro junto con su número de identificación en el sistema y sus coordenadas geográficas de ubicación, siguiendo el formato:

ESTACION,ESTACION\_ID,LATITUD,LONGITUD

Abra el archivo manualmente e inspeccione el formato de los datos usted mismo para asegurarse que el programa los cargará correctamente.

### **metro.log**

Este archivo contiene el registro cronológico de cada entrada y salida de usuarios al metro. Cada línea del archivo representa uno de estos eventos en el siguiente formato:

ESTACION\_ID ID\_USUARIO TIEMPO\_EVENTO TIPO\_EVENTO

El tipo de evento indica si fue de entrada o salida. Abra el archivo manualmente e inspeccione el formato de los datos usted mismo para asegurarse que el programa los cargará correctamente.

Tenga en cuenta que cada usuario debe aparecer al menos dos veces en el archivo, una al entrar y otra al salir. Sin embargo, puede haber varios viajes de un mismo usuario, haciendo que éste aparezca siempre un número par de veces en el archivo.

#### 1.4. Estructura del programa a desarrollar

El programa se compone de los siguientes archivos fuente (módulos):

- **system.py**: Definición de las funciones que permiten generar las estadísticas del menú 1.
- **stations.py**: Definición de las funciones que permiten generar las estadísticas del menú 2.
- **main.py**: Programa principal donde se implementa el menú y la invocación de las funciones que se encuentran en los otros módulos.

El programa debe ser desarrollado utilizando funciones. Parte importante del trabajo será el de definir qué partes del programa se implementarán en funciones. Guíese por la estructura del menú y los archivos fuente mencionados para decidir sobre qué funciones deben crearse.

Solamente la función `stdistance()` se encuentra ya implementada, en el archivo fuente `system.py`. Esta función se encarga de calcular la distancia aproximada entre dos estaciones, dadas sus coordenadas geográficas. Lea con cuidado la especificación de la función para que la pueda utilizar correctamente cuando la necesite.

#### 1.5. Organización de datos

Además de otros tipos de datos, el programa deberá hacer uso de diccionarios. Al inicio, se espera que el programa cargue los datos de los archivos en diccionarios con la siguiente estructura:

Para `stations.info`:

```
{ ESTACION_ID: (ESTACION, LATITUD, LONGITUD) }
```

Para `metro.log`:

```
{ ID_USUARIO: [ (EST_INGRESO, HORA_INGRESO, EST_SALIDA, HORA_SALIDA), (...), ...] }
```

Adicionalmente, para generar las estadísticas requeridas, deberá construir algunos diccionarios adicionales a partir de la información anterior. Uno que relacione las horas con la cantidad de viajes que se hacen durante esa hora:

```
{ HORA: CANT_VIAJES }
```

Y otro que relacione la cantidad de viajes que se hacen desde cada estación:

```
{ ESTACION_ID: CANT_VIAJES }
```

#### 1.6. Opcional

Agregue al menos dos de las siguientes opciones a los correspondientes menús del programa:

Al menú 1:

- Número promedio de viajes de un usuario por día
- Los 5 trayectos más populares en el sistema

Al menú 2:

- Cantidad de usuarios que ingresaron discriminado por hora
- Cantidad de usuarios que salieron discriminado por hora

### **3. Evaluación**

La evaluación se basará en los códigos enviados y una sustentación oral sobre los temas de la práctica. Además, se tendrá en cuenta una bonificación para quien haga la parte opcional.