

# Introducción a la Programación

Nahuel Díaz

Primer semestre de 2022

Clase práctica 11: Tuplas, conjuntos y diccionarios

# Ejercicio 1

Se tienen listas con los datos de personas que viven en localidades de la provincia de Formosa. Cada persona se representa como una tripla (Nombre, DNI, Edad) y en cada lista puede haber personas con el mismo nombre.

Se desea tomar de a dos listas y obtener una lista de nombres ordenada alfabéticamente y sin repetidos que cumpla con las siguientes restricciones:

- ▶ Cada nombre aparece en ambas listas de personas.
- ▶ Cada nombre se corresponde con al menos una persona mayor de 80 años en cada lista.

Resolver el problema usando conjuntos y tuplas. Verificar la implementación mediante casos de test<sup>\*</sup>.

---

<sup>\*</sup> Pueden usar la función `assertListEqual` de la biblioteca `unittest`. Ver <https://docs.python.org/3/library/unittest.html>.

## Ejercicio 1 (cont.)

Por ejemplo, si tuviéramos las siguientes listas de personas:

```
clorinda = [("Dorotea", 14581, 82), ("Azucena", 27652, 75),  
            ("Dorotea", 3245, 89), ("Felisberto", 987, 90)]  
  
pirane = [("Azucena", 8613, 87), ("Dorotea", 5821, 88),  
          ("Felisberto", 2854, 89), ("Luis", 99899, 35)]  
  
mosconi = [("Dorotea", 42939, 67), ("Luis", 1150, 90)]
```

Para clorinda y pirane, la lista resultante debería ser  
["Dorotea", "Felisberto"].

Para clorinda y mosconi, la lista resultante sería una lista vacía.

Para pirane y mosconi, la lista resultante sería una lista vacía.

## Ejercicio 2

En este caso obtuvimos una lista de personas en las cuales podría haber DNIs mellizos (personas distintas con igual número de DNI). Por lo tanto, se quiere obtener un diccionario en donde las claves sean DNIs mellizos y el valor asociado a dicha clave sea una lista de tuplas (Nombre, Edad) de las personas con el número de DNI de la clave. Verificar la implementación mediante casos de test\*.

Por ejemplo, si tuviéramos las siguientes listas de personas:

```
personas = [("Maria", 123456, 25),  
            ("Ahmed", 6841635, 81),  
            ("Jairo", 123456, 65),  
            ("Matias", 355128, 90),  
            ("Tomas", 355120, 8),  
            ("Matias", 355128, 48)]
```

El diccionario resultante debería ser

```
{123456: [('Maria', 25), ('Jairo', 65)],  
 355128: [('Matias', 90), ('Matias', 48)]}
```

---

\* Pueden usar la función `assertDictEqual` de la biblioteca `unittest`.

# A modo de cierre

Hoy vimos ejemplos de problemas en los cuales:

- ▶ usamos los tipos de datos tuplas, conjuntos y diccionarios;
- ▶ compusimos varios tipos de datos para obtener, por ejemplo, listas de tuplas o diccionarios con listas de tuplas como valores;
- ▶ la diversidad de soluciones posibles cada vez es mayor al contar con más variedad de herramientas.

Con lo visto, ya pueden resolver la Guía de Ejercicios 8 completa.