

# NIVEL 3

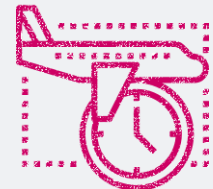
---

## EJERCICIOS





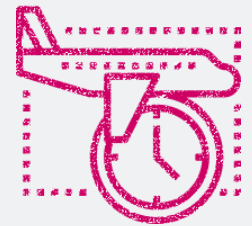
1. Escribe una función que reciba por parámetro el código de un aeropuerto y el diccionario de vuelos y retorne una lista con los códigos de todos los vuelos que parten de ese aeropuerto
2. Escribe una función que reciba por parámetro el código de un vuelo y el diccionario de vuelos y retorne un diccionario con la información del mismo. Si el vuelo no existe, la función retorna None
3. Escribe una función que reciba por parámetro el código IATA de una aerolínea y el diccionario de vuelos y retorne el código del vuelo más largo de dicha aerolínea



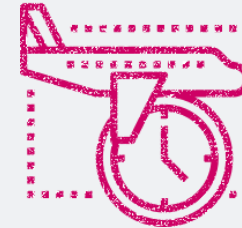
4. Escribe un programa que reciba por parámetro un diccionario de vuelos, un aeropuerto de origen y un aeropuerto destino y muestre al usuario:
- ✓ Todos los vuelos directos que van del aeropuerto de origen al aeropuerto de destino
  - ✓ Todos los vuelos con una escala que van del aeropuerto de origen al aeropuerto de destino

Nota: Antes de escribir el programa:

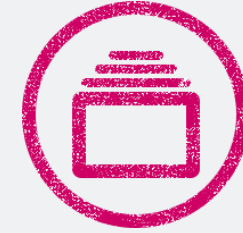
- Plantea en palabras la estrategia de solución y discútela con un compañero@
- Cuando hayan llegado a un acuerdo, definan cuántas y qué funciones necesitan para resolver el problema
- Ahora si, empiecen a programar



5. Escribe una función que retorne el aeropuerto más visitado, teniendo en cuenta tanto los vuelos que salen como los que llegan
6. Escribe una función que reciba por parámetro un diccionario de vuelos y retorne la aerolínea con mas vuelos



## EJERCICIO #6



Vamos a resolver este problema de 3 maneras diferentes:

- ✓ **Con doble recorrido y sin ninguna estructura de datos auxiliar:** se recorren uno a uno todos los vuelos del diccionario. Luego, **para cada uno de los vuelos**, se hace otro recorrido en el cual se cuenta cuántas veces aparece (en el diccionario de vuelos) la aerolínea del **vuelo**. Si ese número de veces es el mayor hasta el momento, se guarda en una variable
- ✓ **Construyendo un histograma:** primero se extraen (puede ser en una lista), las diferentes aerolíneas del diccionario. Luego, **para cada aerolínea** de la lista anterior, se cuenta cuántas veces aparece (en el diccionario de vuelos) y se guarda esta información en una estructura de datos auxiliar (un diccionario, en el cual las llaves son las aerolíneas y los valores la cantidad de veces que aparece cada aerolínea). Por último, se recorre este último diccionario buscando cuál es la aerolínea que aparece un mayor número de veces
- ✓ **Con un solo recorrido utilizando una estructura de datos auxiliar:** piense cómo podría resolver este problema recorriendo una sola vez el diccionario de vuelos

# SOLUCIÓN CON DOBLE RECORRIDO

Si el diccionario de vuelos contiene  $n$  vuelos, esta solución itera:  $n^2$  veces

**Es muy ineficiente !!!**

```
def doble_recorrido(vuelos:dict)->str:
    #Se declaran las variables donde se van a guardar
    #la aerolínea con más vuelos y el número de vuelos de
    #dicha aerolínea
    aerolinea_con_mas_vuelos = None
    mas_vuelos = 0

    #Ciclo externo que recorre todos los vuelos del
    #diccionario
    for cada_vuelo in vuelos:
        #De cada vuelo, se obtiene la aerolínea
        aero = vuelos[cada_vuelo]["aerolinea"]

        #Ciclo interno que recorre nuevamente todos los
        #vuelos del diccionario, contando cuántas veces
        #aparece la aerolínea aero (es decir, cuántos
        #vuelos tiene dicha aerolínea)
        veces = 0
        for cada_vuelo in vuelos:
            if (vuelos[cada_vuelo]["aerolinea"] == aero):
                veces+=1

            #Si el número de veces que aparece la aerolínea aero es
            #mayor que el número de vuelos de la aerolínea con más vuelos
            #que se ha encontrado hasta el momento,
            #se reemplazan las variables
            if veces > mas_vuelos:
                mas_vuelos = veces
                aerolinea_con_mas_vuelos = aero

    #Al final, se retorna la aerolínea con más vuelos
    return aerolinea_con_mas_vuelos
```



# SOLUCIÓN CON HISTOGRAMA

Si el diccionario de vuelos contiene  $n$  vuelos y  $x$  aerolíneas diferentes, esta solución itera:

$n + (n * x) + x$  veces

**Es mucho más eficiente que la solución anterior !!!**

```
def histograma(vuelos:dict)->str:
    #Se construye la lista de aerolíneas
    todas_aerolineas = []
    for cada_vuelo in vuelos:
        aero = vuelos[cada_vuelo]["aerolinea"]
        if aero not in todas_aerolineas:
            todas_aerolineas.append(aero)

    #Se construye el histograma: diccionario de aerolíneas (llaves) con
    #las veces que aparece (valores)
    diccionario_histograma = {}
    #Ciclo que recorre todas las aerolíneas de la lista anterior
    #contando cuántas veces aparece la aerolínea en el diccionario de
    #vuelos (es decir, calculando cuántos vuelos tiene)
    for cada_aerolinea in todas_aerolineas:
        veces = 0
        for cada_vuelo in vuelos:
            if (vuelos[cada_vuelo]["aerolinea"] == cada_aerolinea):
                veces+=1
        diccionario_histograma[cada_aerolinea] = veces

    #Ciclo que recorre el diccionario con el histograma y busca
    #cuál es la aerolínea con mayor frecuencia
    aerolinea_con_mas_vuelos = None
    mas_vuelos = 0
    for cada_aerolinea in diccionario_histograma.keys():
        veces = diccionario_histograma[cada_aerolinea]
        if veces > mas_vuelos:
            mas_vuelos = veces
            aerolinea_con_mas_vuelos = cada_aerolinea

    #Al final, se retorna la aerolínea con más vuelos
    return aerolinea_con_mas_vuelos
```

# SOLUCIÓN CON UN SOLO CICLO

Si el diccionario de vuelos contiene **n** vuelos y **x** aerolíneas diferentes, esta solución itera: **n** veces

**Es súper eficiente !!!**

```
def un_solo_recorrido(vuelos:dict)->str:
    #Se declaran las variables donde se van a guardar
    #la aerolínea con más vuelos y el número de vuelos de
    #dicha aerolínea
    aerolinea_con_mas_vuelos = None
    mas_vuelos = 0

    #Se construye un diccionario auxiliar en el cual
    #se van a guardar las aerolíneas (llaves) y la cantidad de
    #vuelos de cada una (valores)
    diccionario_conteo = {}

    #Se recorre el diccionario de vuelos
    for cada_vuelo in vuelos:
        #Se obtiene la aerolínea de cada vuelo
        aero = vuelos[cada_vuelo]["aerolinea"]
        #Se busca cuántas veces aparece esta aerolínea en
        #el diccionario de conteo que estamos armando.
        #Se usa la función get para esto.
        veces = diccionario_conteo.get(aero,0)
        #Se actualiza el diccionario de conteo con la cantidad
        #de veces incrementada en 1
        diccionario_conteo[aero] = veces + 1
        #Y de una vez se compara este valor con el valor máximo
        #que llevamos hasta el momento
        if diccionario_conteo[aero] > mas_vuelos:
            mas_vuelos = diccionario_conteo[aero]
            aerolinea_con_mas_vuelos = aero

    #Al final, se retorna la aerolínea con más vuelos
    return aerolinea_con_mas_vuelos
```