

EJ2-Diccionarios, Listas y Tuplas

1. DICCIONARIOS

Crea un diccionario con las asignaturas:

CLAVE	NOMBRE
SIGE	Sistemas de Gestión Empresarial
PROM	Programación Multimedia
PROS	Programación servicios y procesos
ADAT	Acceso a datos

El programa debe hacer:

- 1.1 Crear Diccionario de Asignaturas
- 1.2 Añadir Asignatura DEIN
- 1.3 Visualizar el valor del diccionario accediendo con la clave SIGE
- 1.4 Listar las claves y los valores
- 1.5 Comprobar la existencia de varias claves
- 1.6 Eliminar SIGE
- 1.7 Buscar DEIN y si existe visualizar su valor y eliminarla.

Ejemplo de ejecución:

```
ASIGNATURAS: {'PROS': 'Programacion Servicios y Procesos', 'PROM':  
'Programacion Multimedia', 'ADAT': 'Acceso a Datos', 'SIGE': 'Sistemas  
de Gestion Empresarial'}
```

```
AÑADIR ASIGNATURA DEIN
```

```
ASIGNATURAS TRAS AÑADIR: {'PROS': 'Programacion Servicios y Procesos',  
'PROM': 'Programacion Multimedia', 'ADAT': 'Acceso a Datos', 'SIGE':  
'Sistemas de Gestion Empresarial', 'DEIN': 'Desarrollo de Interfaces'}
```

```
Valor de SIGE:  Sistemas de Gestion Empresarial
```

```
LISTADOS
```

```
Lista de las claves:  ['PROS', 'PROM', 'ADAT', 'SIGE', 'DEIN']
```

```
Lista de los valores: ['Programacion Servicios y Procesos',  
'Programacion Multimedia', 'Acceso a Datos', 'Sistemas de Gestion  
Empresarial', 'Desarrollo de Interfaces']
```

```
EXISTE???
```

```
Existe la clave 1? False
```

```
Existe la clave ADAT? True
```

```
Eliminar SIGE
```

```
Claves tras eliminar SIGE: ['PROS', 'PROM', 'ADAT', 'DEIN']
```

```
Busca una función que elimine DEIN y nos devuelva su valor:  
Desarrollo de Interfaces
```

Claves tras eliminar DEIN: ['PROS', 'PROM', 'ADAT']

Busca una función que busque en el diccionario una clave y si no existe devuelva No existe

Ejemplo1: búsqueda de DEIN

No existe

Ejemplo2: búsqueda de ADAT

Acceso a Datos

2. LISTAS

Programa que tiene 3 listas:

- Alumnado (3 personas) que contiene nombres.
- Notas_SIGE (3 notas)
- Notas_ADAT (3 notas)

El programa debe hacer:

2.1 Sacará el listado de las listas por posición, es decir, a quien esté en la posición 0 de Alumnado le corresponden las notas de SIGE en posición 0.

2.2 Posteriormente incluirá 3 nuevos alumnos y sus notas.

2.3 Sumará 0,5 a la nota solamente en el caso de que la nota sea mayor de 5

2.4 Modificará la nota de uno de ellos en ADAT buscándola por el nombre en la lista de Alumnado, cogiendo su índice y accediendo con él a la lista de ADAT

Ejemplo de ejecución:

LISTADO NOTAS

Aitor
- SIGE: 5
- ADAT: 4

Amaia
- SIGE: 3
- ADAT: 9

Mikel
- SIGE: 8
- ADAT: 1

INCLUIDOS NUEVOS ALUMNOS:

- Alumnado actual ['Aitor', 'Amaia', 'Mikel', 'Ainara', 'Juanjo', 'Asier']
- SIGE: [5, 3, 8, 4, 6, 2]
- ADAT: [4, 9, 1, 7, 8, 4]

CALCULAR suma de 0,5 de Actitud en SIGE si la nota actual es >5
Notas SIGE: [5, 3, 8, 4, 6, 2]
+0.5 en aprobados: [5.5, 8.5, 6.5]

MODIFICAR nota de ASIER en ADAT

- Nota actual 4
- Asignacion nueva 6

3. TUPLAS

Construye una tupla de la lista del ejercicio anterior y visualiza el siguiente resultado:

```
-----  
                        TUPLAS  
-----
```

```
TODAS LAS NOTAS  
( 'Aitor', 'Amaia', 'Mikel', 'Ainara', 'Juanjo', 'Asier', 'Aitor',  
'Amaia', 'Mikel', 'Ainara', 'Juanjo', 'Asier')  
SIGE: [5, 3, 8, 4, 6, 2] ADAT [4, 9, 1, 7, 8, 6]
```

```
NOTAS SIGE de los 3 primeros alumnos  
( 'Aitor', 'Amaia', 'Mikel')  
[5, 3, 8]
```

```
NOTAS SIGE de los 3 ultimos alumnos  
( 'Ainara', 'Juanjo', 'Asier')  
[4, 6, 2]
```