



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### Ingeniero en computación

**Materia:** Lenguaje de Programación Python / Clave 36276

**Alumno:** Héctor Daniel Camacho López

**Matrícula:** 372239

**Maestro:** Pedro Núñez Yépiz

**Actividad No. :** 9

**Tema - Unidad :** LISTAS Y DICCIONARIOS

**Ensenada Baja California a 16 de Octubre del 2023**



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 1. INTRODUCCIÓN

En la práctica se verán los diccionarios, se explorarán sus características, las utilidades que poseen por sobre otros tipos de datos o estructuras, y las maneras de utilizar las funciones que pueden modificar y crear los diccionarios. Los diccionarios son una estructura de datos poderosa y versátil que permite almacenar y organizar elementos en pares de clave-valor, estos dos últimos siendo pieza clave para el entendimiento de los diccionarios, pues a la hora de llenar o imprimirlos es necesario observar si se desea trabajar con clave-valor, o con valor-clave. De esta manera, los diccionarios son herramientas útiles para manipular grandes cantidades de datos de una manera versátil, y que a su vez tienen una simplicidad para entenderse que los hace aún más convenientes a la hora de realizar un programa. En esta práctica se explorarán estos puntos que hacen ver a los diccionarios como algo muy útil y conveniente en el almacenamiento de datos.

### 2. COMPETENCIA

Se aprenderá a crear y utilizar diccionarios para almacenar distintos datos generados de manera aleatoria con las funciones de la librería random y las funciones que sirven para modificar los diccionarios.



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 3. FUNDAMENTOS

Un Diccionario es una estructura de datos y un tipo de dato en Python con características especiales que nos permite almacenar cualquier tipo de valor como enteros, cadenas, listas e incluso otras funciones. Estos diccionarios nos permiten además identificar cada elemento por una clave (Key). **Para definir un diccionario**, se encierra el listado de valores entre llaves {}. Las parejas de clave y valor se separan con comas, y la clave y el valor se separan con dos puntos: **Los diccionarios de datos en Python** son una estructura de datos poderosa y versátil que permite almacenar y organizar elementos en pares de clave-valor. Cada elemento en un diccionario está asociado con una clave única que actúa como identificador y se utiliza para acceder a su valor correspondiente.

Las listas son un tipo de dato en Python que permite almacenar una colección de elementos en un solo lugar. Esto significa que podemos guardar varios elementos en una lista y acceder a ellos de manera fácil y ordenada. Las listas son uno de los tipos de datos más utilizados en Python, ya que son muy flexibles y se pueden utilizar para una amplia variedad de aplicaciones. Las listas son una estructura de datos en Python que nos permiten almacenar una colección ordenada de elementos. Estos elementos pueden ser de diferentes tipos, como números, cadenas de texto, booleanos, y hasta otras listas. Una lista es similar a una matriz o vector en otros lenguajes de programación. **Las listas son mutables**, lo que significa que podemos cambiar su contenido. Las listas pueden contener elementos de diferentes tipos. **Los elementos de una lista** se almacenan en orden y se pueden acceder utilizando índices. **Las listas son una estructura de datos flexible**, lo que significa que podemos agregar, eliminar o modificar elementos en cualquier momento.

Link de información:

[https://github.com/pedroypiz/pythonHOY/blob/main/10\\_LISTAS\\_EN\\_PYTHON.ipynb](https://github.com/pedroypiz/pythonHOY/blob/main/10_LISTAS_EN_PYTHON.ipynb)

[https://github.com/pedroypiz/pythonHOY/blob/main/14\\_DICCIONARIOS\\_EN\\_PYTHON.ipynb](https://github.com/pedroypiz/pythonHOY/blob/main/14_DICCIONARIOS_EN_PYTHON.ipynb)

### 4. PROCEDIMIENTO

**Realiza un programa en python que utilice el siguiente menú:**

MENU

- 1.- Crear Diccionario
- 2.- Imprimir Diccionario
- 0.- Salir

**CREAR DICCIONARIO:** función que retorne un diccionario con los datos de básicos de un alumno, donde se pide {id, nombre (1 O 2), apaterno, apmaterno, edad, sexo}, El diccionario se debe generar automáticamente a partir de listas.

**IMPRIMIR DICCIONARIO:** Función que reciba como parámetro el diccionario, y le dé una salida mostrando la información en forma de registro que se generó con el diccionario



# Universidad Autónoma de Baja California

## Facultad de Ingeniería Arquitectura y Diseño

### 5. RESULTADOS Y CONCLUSIONES

Los diccionarios son utilizados para almacenar información, en este caso registros aleatorios que se generaban con nombres, apellidos, edad, sexo e id, simulando personas registradas en el diccionario. Las listas van muy de la mano con los diccionarios, pues estas pueden caber dentro de uno y almacenarse para manipularse posteriormente. Se mencionó la diferencia entre 3 estructuras de datos, los cuales son set, lista y tupla, y tienen diferentes tamaños (en bits) que los hacen más o menos útiles a la hora de cuidar el espacio a la hora de programar. Otro aspecto importante mencionado, es que las librerías normalmente se importan en su totalidad, es decir, con todas sus funciones, pero es recomendable no hacerlo y en su lugar solo importar la o las funciones que se vayan a utilizar. Otra cosa importante, es la manera de organizar los datos de una manera más limpia y ordenada, y es que los diccionarios cuentan con muchas funciones que los manipulan a conveniencia del programador, que van desde simplemente llenar un diccionario o una sección de este, hasta anidar los diccionarios para crear uno solo. El uso de diccionarios es sencillo, y es solo cuestión de prestar atención (en este caso) a los parámetros que se les daban a las funciones y lo que regresaba, para que al momento de imprimir las opciones del menú y el usuario escoja no hubiera errores o no se pasaran los diccionarios correctamente.

### 6. ANEXOS

Link a PDF con código:

<https://drive.google.com/drive/folders/1frAVxE4Lp4cylgM0tCQifU2PBpieqOnk>

### 7. REFERENCIAS

#### **Diseño de algoritmos y su codificación en lenguaje C**

Corona, M.A. y Ancona, M.A. (2011)..  
España: McGraw-Hill.  
ISBN: 9786071505712

#### **Programación estructurada a fondo: implementación de algoritmos en C**

:Pearson Educación. Sznajdleder, P. A. (2017)..  
Buenos Aires, Argentina: Alfaomega

#### **Como programar en C/C++**

H.M. Deitel/ P.J. Deitel  
Segunda edición  
Editorial: Prentice Hall.  
ISBN: 9688804711

#### **Programación en C. Metodología, estructura de datos y objetos**

Joyanes, L. y Zahonero, I. (2001)..  
España: McGraw-Hill.  
ISBN: 8448130138