



# **Universidad Autónoma de Baja California**

## **Facultad de Ingeniería, Arquitectura y Diseño**

Ingeniero en Computación

### **Asignatura:**

Lenguaje de Programación Python

### **Actividad 12:**

Diccionarios

**Brayan Arturo Rocha Meneses**

**Matricula:**

371049

**Ensenada Baja California 8 de Noviembre del 2023**

## **Introducción:**

Las listas y diccionarios son estructuras fundamentales en el lenguaje de programación Python, proporcionando a los desarrolladores herramientas poderosas y flexibles para manipular datos de manera eficiente. Su versatilidad y capacidad para almacenar, organizar y manipular información de diversas maneras.

## **Competencia:**

Las listas como los diccionarios son tipos de datos compuestos en Python, tienen diferencias significativas en términos de estructura y funcionalidad. Las listas son secuencias ordenadas de elementos, accesibles mediante índices numéricos, mientras que los diccionarios son colecciones no ordenadas de pares clave-valor, proporcionando un acceso rápido a los elementos a través de claves únicas.

La elección entre listas y diccionarios depende del contexto y de los requisitos específicos del problema. Las listas son ideales para datos ordenados, como secuencias de números o texto, mientras que los diccionarios son más apropiados cuando se necesita un acceso rápido a los elementos mediante identificadores significativos.

## Fundamentos:

### Listas:

**Mutabilidad:** Las listas son estructuras de datos mutables, lo que significa que se pueden modificar después de su creación. Esto permite realizar operaciones como agregar, eliminar o cambiar elementos en la lista.

**Métodos Incorporados:** Las listas ofrecen una amplia gama de métodos integrados, como `append()`, `extend()`, `insert()`, y `remove()`, que facilitan la manipulación de datos de manera eficiente.

Iteración: Se pueden recorrer fácilmente mediante bucles `for`, lo que simplifica la aplicación de operaciones a cada elemento de la lista.

### Diccionarios:

**Eficiencia en la Búsqueda:** La eficiencia de búsqueda en diccionarios es constante, lo que significa que el tiempo requerido para acceder a un valor a través de una clave no depende del tamaño del diccionario. Esto los hace ideales para operaciones de búsqueda.

**Keys y Values:** Los métodos `keys()` y `values()` permiten acceder a las claves y valores del diccionario, respectivamente, proporcionando flexibilidad al trabajar con diferentes aspectos de la estructura.

**Comprensiones de Diccionario:** Similar a las listas, los diccionarios admiten comprensiones que permiten construir diccionarios de manera concisa mediante expresiones compactas.

### Impresión Bonita de Diccionarios:

Además de utilizar `json.dumps` con la opción `indent`, se pueden explorar otras técnicas para imprimir de manera estética los diccionarios:

```
mi_diccionario = {"clave1": 10, "clave2": "valor", "clave3": [1, 2, 3]}
```

Utilizar un bucle para imprimir línea por línea

```
for key, value in mi_diccionario.items():
```

```
    print(f"{key}: {value}")
```

Utilizar el formato f-string para imprimir

```
print("{}")
```

```
for key, value in mi_diccionario.items():
```

```
    print(f" '{key}': {value},")
```

```
print("{}")
```

## Procedimiento:

### MENU

- 1.- Agregar (automatico 10)
- 2.- Eliminar {ID}
- 3.- Imprimir lista ( tabla)
- 4.- Buscar {ID}
- 5.- Buscar {appat} todas las coincidencias
- 6.- Ordenar {ID}
- 7.- Generar archivo {ID} (preguntar nombre del archivo)
  - a) excel
  - b) txt
  - c) cvs
  - d) Markdown
- 8.- Cargar archivo {ID}
- 9.- Imprimir archivo {ID}
- 10.- Borrar Toda la lista {ID}
- 0.- SALIR

**NOTA: Los datos del diccionario son los Datos básicos de un trabajador de una fabrica.**

### NOTA 2: VALIDAR EL PROGRAMA 100%

Preguntar siempre si esta seguro eliminar, No ordenar si ya esta Ordenado, decir si lista vacía, no existe o mostrar si estala búsqueda, No se permiten ID repetidos

## **Conclusiones:**

Al profundizar en los fundamentos de listas y diccionarios en Python, se destaca la versatilidad y la eficiencia que ofrecen para diferentes tareas de programación. Las listas son adecuadas para secuencias ordenadas y manipulación flexible, mientras que los diccionarios destacan en búsquedas eficientes y asociaciones clave-valor. La impresión estética de diccionarios no solo mejora la legibilidad sino que también contribuye a un código más mantenible.

**Anexo:**

<https://github.com/BrayanArM98/Python>