



## PROCESO DIRECCIÓN DE FORMACIÓN PROFESIONAL INTEGRAL FORMATO GUÍA DE APRENDIZAJE

### 1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE\_

- **DENOMINACIÓN DEL PROGRAMA DE FORMACIÓN:**  
ANALISIS EXPLORATORIO DE DATOS EN PYTHON
- **CÓDIGO DEL PROGRAMA DE FORMACIÓN:** 22810012
- **NOMBRE DEL PROYECTO:** N/A
- **FASE DEL PROYECTO:** N/A
- **ACTIVIDAD DE PROYECTO:** N/A
- **COMPETENCIA:**  
Sistematizar datos masivos de acuerdo con métodos de analítica y herramientas tecnológicas.
- **RESULTADOS DE APRENDIZAJE ALCANZAR:**  
Definir el objetivo del análisis a partir del planteamiento de preguntas que permitan conocer los datos recopilados.
- **DURACIÓN DE LA GUÍA:** 12 horas.

### 1. PRESENTACIÓN

#### Aprendiz SENA:

Un diccionario en Python es una estructura de datos que permite almacenar cualquier tipo de información, desde cadenas de texto o caracteres hasta números enteros, con decimales, listas e incluso otros diccionarios. Al igual que sucede con un diccionario de lengua, los datos se encuentran ordenados utilizando una clave única para cada uno de ellos, lo que permite localizar cada uno de los datos de una forma muy rápida.

#### Diccionarios en Python

Los diccionarios Python son mutables, esto quiere decir que no tienen un tamaño predefinido y que su contenido aumenta o disminuye según las necesidades de la aplicación. Todos los datos son también modificables, es decir, se puede añadir, modificar, eliminar y consultar todos los datos de una manera sencilla y rápida.

Los valores de un diccionario se guardan utilizando un par de valores que siempre van enlazados. Una es denominada como **Key o Clave**, que es la que nos permite encontrar un dato dentro del diccionario. Cada clave está acompañada por el dato o valor al que representa.

Para definir un diccionario deberemos crear una variable utilizando el juego de llaves {}.

Si queremos definir una estructura de datos inicial de forma manual, deberemos usar las mismas llaves y rellenar el par de valores de los que hablamos anteriormente. El primer valor corresponderá con la clave, que tendrá que ir entre comillas simples en caso de ser una cadena, y el segundo, que será el propio valor que queremos almacenar.



```
1 diccionario = {'nombre':'Antonio'}  
2 #El primer valor que se ve se le llama key o clave  
3 #El segundo valor se le conoce como el dato que almacena esa palabra clave
```

```
1 #Obviamente se le puede agregar mas valores y mas datos, separandolos por comas  
2  
3 diccionario = {'nombre':'Antonio','apellido':'Lopez','edad':27}  
4  
5 #El ultimo dato al ser tipo numerico no es necesario colocarlo en comillas simples
```

### Palabras claves de diccionario

**Tuplas:** Las tuplas en Python son un tipo o estructura de datos que permite almacenar datos de una manera muy parecida a las listas, con la salvedad de que son inmutables, que quiere decir inmutables, que después de que son declaradas no se le puede seguir haciendo modificaciones, es una especie de diccionario, solo que en este caso no se utilizan las llaves para declararlas "{}" si no se utilizan los paréntesis "()", aquí también funciona de separar los datos simplemente utilizando comas.

```
1 tupla = (1,2,3,4)  
2 print(tupla)
```



## Métodos de los diccionarios

**Clear():** elimina todos los datos de un diccionario. EJEMPLO: `diccionario.clear()`

```
1 diccionario = {'nombre':'Antonio','apellido':'Lopez','edad':27}
2 print(diccionario.clear())
```

**Get(clave):** permite consultar uno de los datos guardados. EJEMPLO: `diccionario.get(nombre)`

```
1 diccionario = {'nombre':'Antonio','apellido':'Lopez','edad':27}
2 print(diccionario.get('nombre'))
3 #Si colocamos la key de este diccionario
4 #Y al momento de imprimirlo se coloca .get
5 #Nos mostrara el dato de esa key en este caso Antonio
```

**pop(clave):** permite eliminar un elemento del diccionario. EJEMPLO: `diccionario.pop(edad)`

```
1 diccionario = {'nombre':'Antonio','apellido':'Lopez','edad':27}
2 print(diccionario.pop('edad'))
3 print(diccionario)
4 # En este caso se imprime lo que vamos a eliminar
5 # en pantalla en este caso con .pop('clave')
6 # se eliminara la edad, luego se imprime el
7 # diccionario sin la clave edad.
```



**SetDefault():** Este método no sirve para poder agregar un campo en el diccionario, es decir, una clave que no tengamos en el diccionario la agrega simplemente con el valor de NONE, si se quiere digitar un valor se separa por una coma la clave y el valor del nuevo campo del diccionario.

```
1  diccionario = {'Nombre':'Antonio', 'Edad':27}
2  print(diccionario.setdefault('telefono',254352453))
3  print(diccionario)
```

**Diccionario['nombre']** = 'Felipe' permite modificar un dato del diccionario es decir se puede modificar cualquier dato que se encuentre dentro del diccionario.

```
1  diccionario = {'nombre':'Antonio','apellido':'Lopez','edad':27}
2  diccionario['nombre'] = 'Camilo'
3  print(diccionario)
4  # En este diccionario se ingreso modifico la key del nombre
5  # para esto se utilizan los corchetes "['key']" para
6  # poder modificar el campo valor del nombre
```

### Utilidad de los diccionarios

Estos diccionarios pueden tener muchas utilidades al momento de realizar nuestro software, como saber cómo están ordenados, que tipo de dato son, y manejar de una mejor manera todos los tipos de datos que se van ingresando en la aplicación que se esté desarrollando.

Igualmente, estos diccionarios en una empresa en momentos reales se pueden utilizar para varios trabajos específicos como:

- **Agenda:** como un diccionario puede guardar objetos o incluso otros diccionarios, es fácil pensar que se trata de una estructura de datos ideal para realizar una agenda telefónica para nuestro móvil.



Se pueden almacenar todos los datos necesarios de una persona, desde su nombre o teléfono hasta su dirección de correo electrónico.

- **Control de stock:** imagina que tienes una tienda y necesitas guardar de forma ordenada todos tus productos. Un diccionario es una estructura de datos adecuada para programar una aplicación de control de stock, que permita manejar de una forma sencilla el inventario de productos de un negocio.
- **Participantes de una carrera:** también podría hacerse un aplicativo para guardar todos los participantes de una carrera popular, ya que podemos agregar participantes, modificar sus datos o eliminar un participante si este se da de baja.

### 3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

#### 3.1 DESCRIPCIÓN DE LA(S) ACTIVIDAD(ES)

Desarrollar las estructuras básicas de los siguientes ejercicios integrando funciones en Python.

- La actividad desarrollada de manera grupal.
- Con ayuda de Python y manual de Consulta desarrollar los diferentes ítems propuestos en la guía de aprendizaje.
- Entregar el resultado, a través de la plataforma Classroom.

#### FASES DE LA ACTIVIDAD

1. Desarrolle las actividades planteadas en las actividades de aprendizaje (Duración 12 horas).
2. Realizar preguntas e inquietudes que puedan surgir
3. Realice los siguientes algoritmos utilizando Python
4. Entregar, a través de la plataforma Classroom.

#### ACTIVIDAD DE APRENDIZAJE:

En este espacio de trabajo se busca identificar e iniciar el aprendizaje incorporando sentencias de control manejadas por el pseudolenguaje o lenguaje de programación. Se incorporan ejercicios prácticos de identificación de variables de entrada y salida, constantes, tipos de datos y comprender el manejo de Diccionarios en Python.



### EJEMPLO:

```
1  #Escribir un programa que guarde en una variable el diccionario
2  #{'Euro':'€', 'Dollar':'$', 'Yen':'¥'}, pregunte al usuario por
3  #una divisa y muestre su símbolo o un mensaje de aviso si la
4  #divisa no está en el diccionario.
5
6  monedas = {'Euro':'€', 'Dollar':'$', 'Yen':'¥'}
7  moneda = input("Introduce una divisa: ")
8  print(monedas.get(moneda.title(), "La divisa no está. "))
9
10 #En la parte de moneda.title() este .title() nos sirve para
11 #colocar la primera letra en mayúscula y el resto en
12 #minúsculas.
```

### EJERCICIOS

1. Escribir un programa que pregunte al usuario su nombre, edad, dirección y teléfono y lo guarde en un diccionario. Después debe mostrar por pantalla el mensaje <nombre> tiene <edad> años, vive en <dirección> y su número de teléfono es <teléfono>.
2. Escribir un programa que guarde en un diccionario los precios de las frutas de la tabla, pregunte al usuario por una fruta, un número de kilos y muestre por pantalla el precio de ese número de kilos de fruta. Si la fruta no está en el diccionario debe mostrar un mensaje informando de ello.

Fruta	Precio
Plátano	1.35
Manzana	0.80
Pera	0.85
Naranja	0.70

3. Escribe un programa Python que pida un número por teclado y que cree un diccionario cuyas claves sean desde el número 1 hasta el número indicado, y los valores sean los cuadrados de las claves.
4. Escribir un programa que permita gestionar la base de datos de clientes de una empresa. Los clientes se guardarán en un diccionario en el que la clave de cada cliente será su NIF, y el valor será otro diccionario con los datos del cliente (nombre, dirección, teléfono, correo, preferente),



donde preferente tendrá el valor True si se trata de un cliente preferente. El programa debe preguntar al usuario por una opción del siguiente menú:

- (1) Añadir cliente,
- (2) Eliminar cliente,
- (3) Mostrar cliente,
- (4) Listar todos los clientes,
- (5) Listar clientes preferentes,
- (6) Terminar.

En función de la opción elegida el programa tendrá que hacer lo siguiente:

- Preguntar los datos del cliente, crear un diccionario con los datos y añadirlo a la base de datos.
- Preguntar por el NIF del cliente y eliminar sus datos de la base de datos.
- Preguntar por el NIF del cliente y mostrar sus datos.
- Mostrar lista de todos los clientes de la base datos con su NIF y nombre.
- Mostrar la lista de clientes preferentes de la base de datos con su NIF y nombre.

5. Crear un programa en Python donde se le pregunte al usuario su nombre su edad y su número de documento, todo esto deberá estar almacenado en un diccionario.
6. En una escuela se necesita un programa el cual gestione la cantidad de personas que entran a cuyo salón, teniendo en cuenta que se debe que tener la información personal de cada estudiante y de los profesores, así mismo, se debe tener un registro del número del salón al cual se le valla a hacer la gestión y que al final se muestre en pantalla.
7. Codifica un programa en Python que nos permita guardar los nombres de los alumnos de una clase y las notas que han obtenido. Cada alumno puede tener distinta cantidad de notas. Guarda la información en un diccionario cuyas claves serán los nombres de los alumnos y los valores serán listas con las notas de cada alumno.  
El programa pedirá el número de alumnos que vamos a introducir, pedirá su nombre e irá pidiendo sus notas hasta que introduzcamos un número negativo. Al final el programa nos mostrará la lista de alumnos y la nota media obtenida por cada uno de ellos. Nota: si se introduce el nombre de un alumno que ya existe el programa nos dará un error.
8. Realizar un algoritmo el cual pregunte el nombre de la persona y la edad, tener en cuenta que al momento de mostrar la edad en pantalla debe mostrar la fecha de nacimiento de dicha persona.
9. Realizar un algoritmo para determinar cuánto ahorrará una persona en un año, si al final de cada mes deposita cantidades variables de dinero; además, se quiere saber cuánto lleva ahorrado cada mes.
10. Conozca si tiene que declarar el impuesto de renta desde el próximo año, según sus ingresos mensuales, si sus ingresos mensuales son mayores de 3.000.000 se le deberá cobrar 250.000 más la implementación de un IVA del 18% de los ingresos mensuales, si los ingresos mensuales son menores a 3.000.000 se le cobrara 200.000 más la implementación del IVA aproximadamente del 10% de sus ingresos mensuales.



11. Realizar un algoritmo el cual sume los ingresos de una empresa mensualmente, teniendo en cuenta que se debe saber de que son las ganancias y se debe pedir al usuario de cuanto dinero total se obtuvo de esa ganancia, al final se deberá saber si las ganancias son mayores a las ganancias del año pasado, entonces imprimir en pantalla que se obtuvieron mas ganancias y hacer la respectiva operación para saber de cuanta diferencia fue la ganancia, si las ganancias son menores a las ganancias del año pasado imprimir en pantalla el faltante de ganancias para completar las ganancias del año pasado. *GANANCIAS: Pedir al usuario que ingrese las ganancias del año pasado.*

**AMBIENTE REQUERIDO:** Fábrica de software

**MATERIALES:**

DEVOLUTIVOS: Pc, Internet, Herramientas ofimáticas, Visual Code o Colaboratory

#### 4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
<b>Evidencias de Desempeño:</b> Entrega de trabajo con el desarrollo de las actividades propuestas en la actividad de aprendizaje: variables, constantes, entrada de datos y salida de información.	Identifica y describe, en un sistema de información dado, los datos de entrada, procesamiento de los datos e información generada, según necesidades del cliente.	Entregar el resultado de la investigación en un informe final, a través de la plataforma Classroom

#### 5. GLOSARIO DE TÉRMINOS

- **Archivos:** Un archivo es una estructura de datos consistente en una secuencia de elementos o componentes Llamados registros, todos del mismo tipo, ya sea simple o estructurado.
- **Constantes:** Una constante es un dato cuyo valor no puede cambiar durante la ejecución del programa. Recibe un valor en el momento de la compilación y este permanece inalterado durante todo el programa.
- **Dato:** Los datos son una representación simbólica (numérica, alfabética, algorítmica) de un atributo o variable cuantitativa referente a una persona, cosa, u objetos.
- **Datos Alfanuméricos:** Es una secuencia de caracteres alfanuméricos que permiten representar valores identificables de forma descriptiva, esto incluye nombre de personas, direcciones, etc.
- **Datos lógicos:** Son aquellos que solo pueden tener dos valores (verdadero o falso) ya que representan el Resultado de una comparación entre otros datos (numéricos y alfanuméricos).
- **Datos numéricos:** Permiten representar valores escalares de forma numérica, esto incluye a los números enteros y los reales. Este tipo de datos permiten realizar operaciones aritméticas comunes.
- **Diagrama de flujo:** Un diagrama de flujo es la representación gráfica de un algoritmo. También se puede decir que es la representación detallada en forma gráfica de cómo deben realizarse los pasos en la computadora para producir resultados.
- **Información:** Es un conjunto organizado de datos, que constituyen un mensaje sobre un determinado ente o fenómeno.





- **Pseudocódigo:** Mezcla de lenguaje de programación y español (o inglés o cualquier otro idioma) que se emplea, dentro de la programación estructurada, para realizar el diseño de un programa. En esencial, el Pseudocódigo se puede definir como un lenguaje de especificaciones de algoritmos
- **Variable:** Es una función que asocia a cada elemento de la población la medición de una característica, particularmente de la característica que se desea observar.

## 6. REFERENTES BIBLIOGRÁFICOS

- Jiménez, J, Alvarado, L, Jiménez, E. (2014). Fundamentos de Programación. Obtenido de [https://www.walfaomegacloud-com.bdigital.sena.edu.co/auth/ip?intended\\_url=https://www-alfaomegacloudcom.bdigital.sena.edu.co/reader/fundamentos-de-programacion-diagramas-de-flujo-diagramas-n-spseudocodigo-y-java?location=20](https://www.walfaomegacloud-com.bdigital.sena.edu.co/auth/ip?intended_url=https://www-alfaomegacloudcom.bdigital.sena.edu.co/reader/fundamentos-de-programacion-diagramas-de-flujo-diagramas-n-spseudocodigo-y-java?location=20)
- Llopis Pascual, F.; Corbí Bellot, A.; Llorens Larg. (s.f.). Fundamentos de programación. Vol. I. Metodología. Obtenido de <https://www.digitaliapublishing-com.bdigital.sena.edu.co/visor/663>
- Wikipedia, «Algoritmo,» Wikipedia Enciclopedia Libre, 25 Febrero 2019. [En línea]. Available: <https://es.wikipedia.org/wiki/Algoritmo>. [Último acceso: 27 Febrero 2019].
- Scribd, «Ejemplo de algoritmos,» Scribd, 18 Octubre 2008. [En línea]. Available: <https://es.scribd.com/doc/7119127/Ejemplos-de-Algoritmos>. [Último acceso: 26 Febrero 2019].
- E. Reyes Corredor, «Aprender a Programar,» Fundación Universitaria San Mateo, 2008. [En línea]. Available: [https://www.researchgate.net/publication/324216652\\_Aprender\\_a\\_programar](https://www.researchgate.net/publication/324216652_Aprender_a_programar). [Último acceso: 26 Febrero 2019].

## 7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
<b>Autor (es)</b>	JHON FREDY CARREÑO PINTO	Aprendiz ADSO	Teleinformática – Centro industrial de Mantenimiento Integral.	Mayo 2023

## 8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
<b>Autor (es)</b>	Enzy Zulay Angarita Bermúdez	Instructora	Teleinformática – Centro industrial de Mantenimiento Integral.	Mayo 2024	Cambio actualización de formato de Guía de aprendizaje. GFPI-F-135 V02