



PROCESO DIRECCIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE 09

- **DENOMINACIÓN DEL PROGRAMA DE FORMACIÓN:**
Análisis y desarrollo de software
 - **CÓDIGO DEL PROGRAMA DE FORMACIÓN:** 228118
 - **NOMBRE DEL PROYECTO:**
Construcción de software integrador de tecnologías orientadas a servicios.
 - **FASE DEL PROYECTO:** Análisis
 - **ACTIVIDAD DE PROYECTO :**
Análisis y comprensión de herramientas tecnológicas necesarias para el manejo de la información en la empresa
 - **COMPETENCIA:**
Establecer requisitos de la solución de software de acuerdo con estándares y procedimiento técnico.
 - **RESULTADOS DE APRENDIZAJE ALCANZAR:**
Recolectar información del software a construir de acuerdo con las necesidades del cliente.
- DURACIÓN DE LA GUÍA:** 12 horas

1. PRESENTACIÓN

Aprendiz SENA:

¿Qué es un diccionario?

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) y valores (values).

- Son conjuntos de datos desordenados (al contrario que las listas, que eran ordenadas)
- Las claves se utilizan para acceder a los elementos y no una posición.

Cada clave del diccionario tiene un valor, que puede ser de cualquier tipo de objeto de Python. Es decir, los diccionarios pueden ser considerados como pares de clave-valor.

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.

Ejemplo:

- `edades = {'Miguel': 22, 'Daniel': 25, 'Bruma': 9}`
- `diccionario = {'nombre' : 'Carlos', 'edad' : 22, 'cursos': ['Python', 'Django', 'JavaScript'] }`



- `animales = {"perro" : "dog", "gato" : "cat", "tiburon" : "shark"}`
`} # Diccionario`

1. Para acceder al elemento de un Diccionario mediante la clave de este elemento, como veremos a continuación:

- `print diccionario['nombre'] #Carlos`
- `print diccionario['edad'] #22`
- `print diccionario['cursos'] #['Python', 'Django', 'JavaScript']`
- `print(animales) # print todo el diccionario`
- `print(animales["perro"]) #print el valor de la clave perro`

2. Para recorrer todo el Diccionario, podemos hacer uso de la estructura for:

- `for key in diccionario:`
`print key, ":", diccionario[key]`

3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 DESCRIPCIÓN DE LA(S) ACTIVIDAD(ES)

Desarrollar las estructuras básicas de los siguientes ejercicios diccionario 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 o Drive misena.

FASES DE LA ACTIVIDAD

1. Desarrolle las actividades planteados 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 o Drive misena.

ACTIVIDAD DE APRENDIZAJE:

1. Identificar la función que devuelva el número de elementos que tiene un diccionario, dar ejemplos
2. Función que permite comparar el número de elementos de diferentes entre dos diccionarios.
- 3.Cuál es la función que permite eliminar algún elemento del diccionario. Dar un ejemplo
- 4.Cuál es la función que permite devolver el número de elementos que tiene el diccionario, ingrese un ejemplo.
5. Función que permite eliminar todos los elementos de un diccionario
6. `print(direccion_datos.values())` # cual es el uso de esta función incorporar a un ejercicio.
7. `print(direccion_datos.items())` # cual es el uso de esta función incorporar a un ejercicio.
8. Función que permite añadir los elementos de un diccionario a otro



EJEMPLO #1:

```
2  '''
3  Colecciones
4  '''
5  # ---> Listas []
6  # ---> Diccionarios {}
7  # ---> Tuplas ()
8
9  '''
10 Declaracion de colecciones
11 '''
12 lista = [] # listas
13 diccionarios = {} # Diccionarios
14
15 numeros = [1,2,3,"hola"] #Lista
16 print(numeros) # print toda la lista
17 print(numeros[0]) # print posicion 0 de la lista
18 print("-----")
19
20 #-----
21         CLAVE  VALOR
22 animales ={"perro" : "dog", "gato" : "cat", "tiburon" : "shark" } # Diccionario
23 print(animales) # print todo el diccionario
24 print(animales["perro"]) #print el valor de la clave perro
```

```
[1, 2, 3, 'hola']
1
-----
{'perro': 'dog', 'gato': 'cat', 'tiburon': 'shark'}
dog
```

EJEMPLO #2:



```
1
2 #Pedimos datos al usuario
3 nombre=str(input("Ingrese nombre: "))
4 apellido=str(input("Ingrese apellido: "))
5 celular=int(input("Ingrese numero celular: "))
6
7 # creamos diccionario
8 agenda ={
9     #Podemos darle valor a las claves con alguna variable
10     CLAVE      VALOR
11     "nombre"   : nombre,
12     "apellido" : apellido,
13     "celular"  : celular
14 }
15
16 #imprimimos
17 print("-----")
18 print(agenda) #Imprimimos todo el diccionario
19 print("-----")
20 print(agenda["apellido"]) #Imprimos el valor de la clave apellido
21
```

```
Ingrese nombre: Maria
Ingrese apellido: Fester
Ingrese numero celular: 3124832718
-----
{'nombre': 'Maria', 'apellido': 'Fester', 'celular': 3124832718}
-----
Fester
```



EJEMPLO #3: AGREGAR Y ELIMINAR ELEMENTOS

```
# AGREGAR Y QUITAR ELEMENTOS DEL DICCIONARIO
```

```
lista=[1,2,3,4] # lista
nombres={ "andres" : [28,5,1994], "Felipe": {"edad": 27, "ciudad": "barranca"}} #diccionario original
print(nombres) # vemos el diccionario original

Clave Valor
nombres["jose"] = lista # aqui agregamos clave jose y su valor una lista
del(nombres["andres"]) # con la funcion del podemos eliminar elementos del diccionario
print("-----")
print(nombres) # aqui vemos el diccionario con cambios ya realizados
```

```
{'andres': [28, 5, 1994], 'Felipe': {'edad': 27, 'ciudad': 'barranca'}} Diccionario original
-----
{'Felipe': {'edad': 27, 'ciudad': 'barranca'}, 'jose': [1, 2, 3, 4]} Diccionario modificado
```

EJEMPLO #4:

```
#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>.
sumap=0
for i in range(3):
    nom=input("ingrese el nombre ")
    edad=int(input("INGRESE LA EDAD "))
    dir=input("ingrese la dirección ")
    telf=input("ingrese el telefono ")
    direccion_datos={"nombre":nom,
                    "edad":edad }
```



```
sumap=sumap+direccion_datos["edad"]
print(direccion_datos)
print(direccion_datos.keys()) # muestra las claves
print(direccion_datos.values()) # muestra los valores
print(direccion_datos.items()) # muestra la pareja de
clave valor
prom=sumap/3
print("promedio ", prom)
for x, y in direccion_datos.items():
    print("Key: ",x, "valor: ", y)
```

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. Escribir un programa que almacene el diccionario con los créditos de las asignaturas de un curso {'Matemáticas': 6, 'Física': 4, 'Química': 5} y después muestre por pantalla los créditos de cada asignatura en el formato <asignatura> tiene <créditos> créditos, donde <asignatura> es cada una de las asignaturas del curso, y <créditos> son sus créditos. Al final debe mostrar también el número total de créditos del curso.
4. Escribir un programa que cree un diccionario simulando una cesta de la compra. El programa debe preguntar el artículo y su precio y añadir el par al diccionario, hasta que el usuario decida terminar. Después se debe mostrar por pantalla la lista de la compra y el coste total, con el siguiente formato.



Lista de la compra	
Artículo 1	Precio
Artículo 2	Precio
Artículo 3	Precio
Total	Coste

5. Escribir un programa que cree un diccionario de traducción español-inglés. El usuario introducirá las palabras en español e inglés separadas por dos puntos, y cada par <palabra>:<traducción> separados por comas. El programa debe crear un diccionario con las palabras y sus traducciones. Después pedirá una frase en español y utilizará el diccionario para traducirla palabra a palabra. Si una palabra no está en el diccionario debe dejarla sin traducir.
6. Escribir un programa que gestione las facturas pendientes de cobro de una empresa. Las facturas se almacenarán en un diccionario donde la clave de cada factura será el número de factura y el valor el coste de la factura. El programa debe preguntar al usuario si quiere añadir una nueva factura, pagar una existente o terminar. Si desea añadir una nueva factura se preguntará por el número de factura y su coste y se añadirá al diccionario. Si se desea pagar una factura se preguntará por el número de factura y se eliminará del diccionario. Después de cada operación el programa debe mostrar por pantalla la cantidad cobrada hasta el momento y la cantidad pendiente de cobro.

AMBIENTE REQUERIDO: Desarrollo 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
Evidencias de Desempeño: 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 o Drive misena.



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 esencia, 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-alfaomegacloud-com.bdigital.sena.edu.co/auth/ip?intended_url=https://www-alfaomegacloud-com.bdigital.sena.edu.co/reader/fundamentos-de-programacion-diagramas-de-flujo-diagramas-n-s-pseudocodigo-y-java?location=20
- Llopis Pascual, F.; Corbí Bellot, A.; Llorens Larg. (s.f.). Fundamentos de programación. Vol. I. Metodología. Obtenido de <https://www-digitalpublishing-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. [Último acceso: 26 Febrero 2019].



7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Enzy Zulay Angarita Bermudez	Instructora	Teleinformática-Centro Industrial de Mantenimiento Integral	Sept 2022

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					