

**ALGORITMOS Y ESTRUCTURAS DE DATOS (TSDS)**

|  |  |
| --- | --- |
|  | |
| ASIGNATURA: | ALGORITMOS Y ESTRUCTURAS DE DATOS |
| PROFESOR:  FECHA: | Ing. Lorena Chulde  20/06/2025 |
| PERÍODO ACADÉMICO: | 2025-A |
|  | |

**TALLER**

(Individual)

|  |
| --- |
|  |
| **TÍTULO:**  **TUPLAS Y DICCIONARIOS**  **ALGORITMOS** |
| **Nombre del estudiante** |
| XXXXXXXXX |
|  |
| Tuplas - Aprende Python |
|  |

**PROPÓSITO DE LA TAREA**

* Recorrer tuplas mediante estructuras cíclicas para leer sus elementos y visualizar sus valores en pantalla.
* Desarrollar algoritmos usando diccionarios
* Aplicar diccionario mediante funciones para implementar un CRUD

**Parte 1: Tuplas**

1. Iterar una tupla

tupla = 1, 2, 3

|  |
| --- |
| print("Claudia Coello")  lista = [1,2,3]  tupla = tuple(lista)  for i in tupla:  print(i) |
|  |

1. Anidar tuplas. Acceder al elemento ‘a’

tupla = 1, 2, ('a', 'b'), 3

|  |
| --- |
| print("Claudia Coello")  tupla = 1, 2, ('a', 'b'), 3  elemento = tupla[2][0]  print(elemento) |
|  |

1. Asignar el valor de una tupla con n elementos a n variables.

tupla = (1, 2, 3)

|  |
| --- |
| print("Claudia Coello")  tupla = (1, 2, 3)  a,b,c = tupla  print(a)  print(b)  print(c) |
|  |

## Métodos tuplas

1. **count(<obj>)**: Contar el numero de veces que se repite un elemento en la tupla

tupla = (1, 1, 1, 3, 5)

|  |
| --- |
| print("Claudia Coello")  tupla = (1, 1, 1, 3, 5)  print(tupla.count(1)) |
|  |

1. Verificar la posición de un elemento:

### index(<obj>[,index]) : permite buscar el ****índice**** del primer elemento que coincida con un valor específico, **a partir de una posición determinada.**

Con un parámetro

tupla = (7, 7, 7, 3, 5)

|  |
| --- |
| print("Claudia Coello")  tupla = (7, 7, 7, 3, 5)  print(tupla.index(5)) |
|  |

1. Con dos parámetros

tupla = (7, 7, 7, 3, 5)

|  |
| --- |
| print("Claudia Coello")  tupla = (7, 7, 7, 3, 5)  print(tupla.index(7,2)) |
|  |

1. Imprimir el apellido y el nombre

nombres = ("Lorena", "Chulde")

|  |
| --- |
| print("Claudia Coello")  nombres = ("Claudia", "Coello")  nombre, apellido = nombres  print(f"{nombre}, {apellido}") |
|  |

1. Imprimir el orden de llegada de los atletas (enumerate)

atletas = ("Lorena Chulde", "Juan Perez", "Maria Mera", "Pedro Robayo")

|  |
| --- |
| print("Claudia Coello")  atletas = ("Lorena Chulde", "Juan Perez", "Maria Mera", "Pedro Robayo")  for atleta in enumerate(atletas, + 1):  print("Orden", atleta[0],"atleta:",atleta[1]) |
|  |

1. Verificar que es un objeto enumerate

atletas = ("Lorena Chulde", "Juan Perez", "Maria Mera", "Pedro Robayo")

posicion = enumerate(atletas)

print(posicion)

Salida:

<enumerate object at 0x000001799C055EE0>

1. Crea una tupla con números, pide un numero por teclado e indica cuantas veces se repite. Usando count

|  |
| --- |
| print("Claudia Coello")  numeros = (5,4,3,2,1,6,45,6,7,8,9,2,6,3,2,1,6,7)  numero = int(input("Ingrese un numero: "))  print(numeros.count(numero)) |
|  |

1. Crea una tupla con números e indica el numero con mayor valor y el que menor tenga. Usando max, min

|  |
| --- |
| print("Claudia Coello")  numeros = (5,4,3,2,1,6,45,6,7,8,9,2,6,3,2,1,6,7)  print(max(numeros))  print(min(numeros)) |
|  |
| print("Claudia Coello")  numeros = (5,4,3,2,1,6,45,6,7,8,9,2,6,3,2,1,6,7)  numMayor = numeros[0]  numMenor = numeros[0]  for i in range(1, len(numeros)):  if numeros[i] > numMayor:  numMayor = numeros[i]  for i in range(1, len(numeros)):  if numeros[i] < numMenor:  numMenor = numeros[i]  print("El mayor es: ", numMayor)  print("El menor es: ", numMenor) |
|  |

1. Desarrolle un algoritmo que cree una dupla
2. Crea una tupla con valores ya predefinidos del 1 al 10, pide un índice por teclado y muestra el valor de la tupla.

|  |
| --- |
| print("Claudia Coello")  tupla = (1,2,3,4,5,6,7,8,9,10)  indice = int(input("Ingrese un indice: "))  print(tupla[indice]) |
|  |

1. Convertir una lista en tupla haciendo uso de al función tuple().

lista = [6, 7, 8, 9, 10]

|  |
| --- |
| print("Claudia Coello")  lista = [6, 7, 8, 9, 10]  listaTupla = tuple(lista)  print(listaTupla) |
|  |

1. Crea una tupla con los meses del año, pide números al usuario, si el numero esta entre 1 y la longitud máxima de la tupla, muestra el contenido de esa posición sino muestra un mensaje de error.

El programa termina cuando el usuario introduce un cero.

|  |
| --- |
|  |
|  |

1. Crea una función que reciba una tupla de números y devuelva la suma de todos sus elementos.

|  |
| --- |
|  |
|  |

1. Crea una función que reciba una tupla de números y devuelva una nueva tupla con el valor mínimo y máximo.

|  |
| --- |
|  |
|  |

1. Crea una función que reciba una tupla y un valor, y retorne cuántas veces aparece ese valor en la tupla.

|  |
| --- |
|  |
|  |

**Parte 2: Diccionarios**

1. Declarar el diccionario con clave y valor de los países y capitales de América Latina

|  |
| --- |
| print("Claudia Coello")  paises = {"Ecuador": "Quito", "Peru":"Lima","Colombia":"Bogota"}  print(paises) |
|  |

1. Acceder a un valor del diccionario de países y capitales.

|  |
| --- |
|  |
|  |

1. Agregar un elemento al diccionario de países y capitales.

|  |
| --- |
|  |
|  |

1. Modificar un elemento del diccionario de países y capitales.

|  |
| --- |
|  |
|  |

1. Eliminar un elemento del diccionario de países y capitales.

|  |
| --- |
|  |
|  |

1. Eliminar un elemento con la función pop

|  |
| --- |
|  |
|  |

1. Acceder un elemento en concreto

|  |
| --- |
|  |
|  |

1. Crear diccionarios con diferentes tipos de datos

|  |
| --- |
|  |
|  |

1. Con valores de tipo lista

|  |
| --- |
|  |
|  |

1. Un diccionario puede contener otro diccionario

|  |
| --- |
|  |
|  |

1. Consultar las claves del diccionario

|  |
| --- |
|  |
|  |

1. Consultar los valores del diccionario

|  |
| --- |
|  |
|  |

1. Consultar la longitud del diccionario

|  |
| --- |
|  |
|  |

1. Recorrer el diccionario con for e imprimir claves

1. Recorrer el diccionario con for e imprimir clave y valor
2. Escribir un programa que guarde en una variable el diccionario {'Euro':'€', 'Dollar':'$', 'Yen':'¥'}, pregunte al usuario por una divisa y muestre su símbolo o un mensaje de aviso si la divisa no está en el diccionario.
3. 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>.

**CRUD CON DICCIONARIOS**

1. Diccionario que permite gestionar notas

Realiza lo siguiente:

* Crear una función para añadir las notas (agregarNota), con los parámetros: nota, título, contenido.
* Crear una función para ver las notas (verNotas)
* Crear una función para editar las notas (editarNota)
* Crear una función para eliminar las notas (eliminarNota)
* Crear una función que muestre el menú

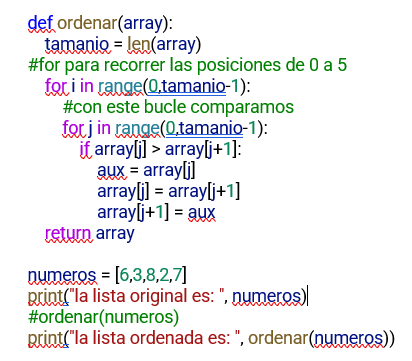
**PARTE 3: Algoritmos**

**Bubble Sort (Burbuja)**

Crear una fn para recorrer una lista.

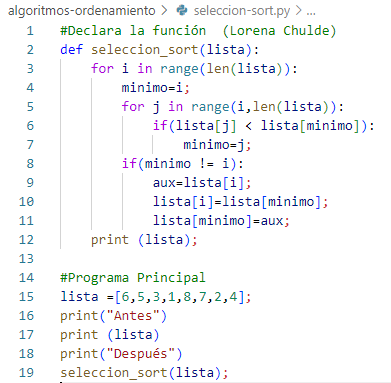
**Método burbuja:** Crear una función para ordenar elementos de un array de menor a mayor.

Cree una función llamada ordenar()



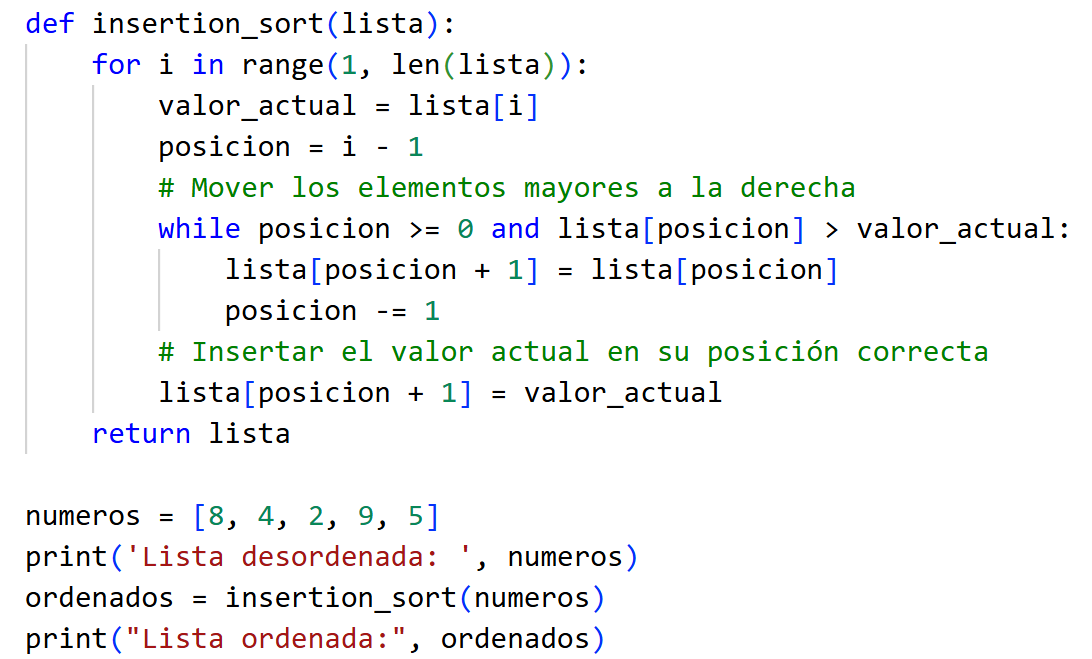
**Selection Sort (Selección)**

* Analizar y explicar paso a paso en qué consiste el Algoritmo de ordenamiento por Selección. Realizar la prueba de escritorio.



**Insertion Sort (Inserción)**

* Recorre la lista de izquierda a derecha.
* En cada paso, **toma un elemento y lo inserta** en la posición correcta dentro de la parte ya ordenada de la lista.



**ENTREGABLES:**

* Una vez culminada tu tarea, capturar las pantallas de la ejecución de cada ejercicio con tus datos y súbela en el apartado del aula virtual.
* Subir los ejercicios al git o al drive y entrega la url de los archivos .py o, a su vez, entregue el archivo.
* Recordar que el nombre del archivo deberá ser: S10\_Taller\_ApellidoNombre

RECURSOS NECESARIOS

* Acceso a Internet.
* Imaginación.
* VSC