



## **ASIGNATURA**

## FUNDAMENTOS DE PROGRAMACIÓN



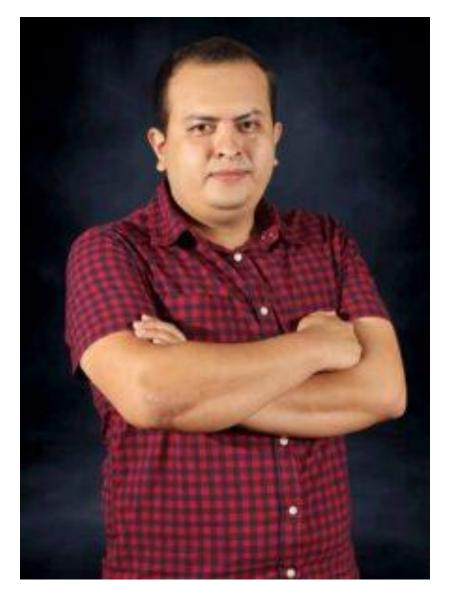
#### Transformamos el mundo desde la Amazonía

Ing. Edwin Gustavo Fernández Sánchez, Mgs.

DOCENTE - PERSONAL ACADÉMICO NO TITULAR OCASIONAL

DIRECTOR DE GESTIÓN DE TECNOLOGÍAS INFORMACIÓN Y COMUNICACIÓN

#### **UNIVERSIDAD ESTATAL AMAZÓNICA**







DESARROLLO DE LA SEMANA 15: DEL LUN. 24 AL DOM. 30 DE MARZO/2025

Resultado de aprendizaje: Utilizar modularización para la simplificación de la estructura de un programa

#### **CONTENIDOS**

**UNIDAD IV : Funciones, Colecciones y Archivos** 

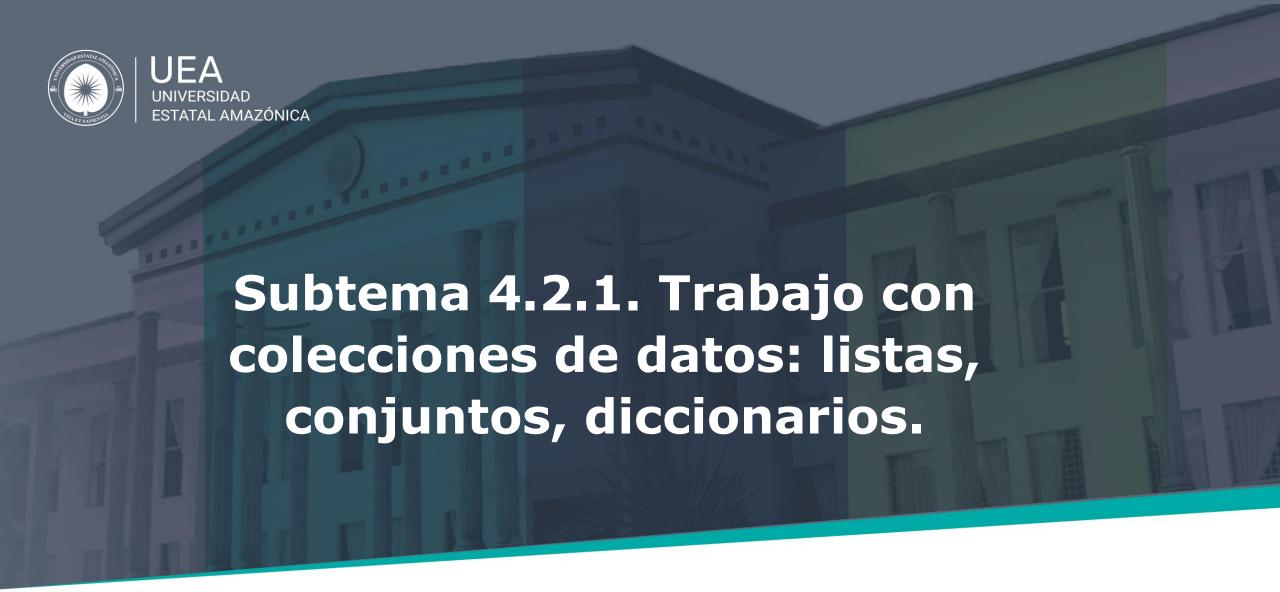
- Tema 4.2: Colecciones y archivos
  - Subtema 4.2.1: Trabajo con colecciones de datos: listas, conjuntos, diccionarios.



# Unidad 4 Funciones, Colecciones y Archivos

**Tema 4.2** 

Colecciones y archivos



**FUNDAMENTOS DE PROGRAMACIÓN** (UEA-L-UFB-026)



## Trabajo con Colecciones de Datos en Python

- En esta sección, nos adentraremos en el emocionante mundo de las colecciones de datos en Python.
- Las colecciones de datos son un componente esencial de la programación, ya que nos permiten gestionar y manipular conjuntos de información de manera eficiente.
- En este subtema, aprenderemos cómo trabajar con tres tipos fundamentales de colecciones en Python: listas, conjuntos y diccionarios.



### Importancia de las Colecciones de Datos en la Programación

- Imagina que estás desarrollando una aplicación de gestión de tareas.
- ¿Cómo almacenarías y accederías a la lista de tareas pendientes de un usuario?
- ¿O qué pasa si estás creando un juego y necesitas llevar un registro de los puntajes de los jugadores?
- En situaciones como estas, las colecciones de datos son tus mejores amigas. Te permiten organizar y manipular información de manera eficaz.



#### Los Tres Pilares de las Colecciones de Datos

- Los tres pilares fundamentales de las colecciones de datos en Python:
  - Listas: Perfectas para almacenar elementos en un orden específico y modificarlos según sea necesario.
  - Conjuntos: Ideales para mantener elementos únicos sin preocuparte por el orden.
  - **Diccionarios**: Excelentes para asociar datos en pares clave-valor y buscar información rápida y eficientemente.
- Operaciones con Listas
  - operaciones comunes en listas, como agregar elementos, eliminar elementos, acceder a elementos por índice y separar listas.



## ¿Qué son las Listas?

- Las listas son colecciones ordenadas de elementos que pueden contener datos de cualquier tipo, como números, cadenas de texto, objetos, e incluso otras listas.
- Se definen utilizando corchetes [] y los elementos dentro de la lista están separados por comas.

## Características Clave de las Listas

- Ordenadas: Los elementos en una lista se almacenan en un orden específico y se pueden acceder por su posición en la lista.
- Modificables: Puedes agregar, eliminar o modificar elementos en una lista de manera dinámica.
- Flexibles: Las listas pueden contener una mezcla de tipos de datos, lo que las hace muy versátiles.



## Ejemplo de Lista

```
python

mi_lista = [1, 2, 3, "cuatro", 5.0]
```

 Aquí, hemos creado una lista llamada mi\_lista que contiene una variedad de tipos de datos, desde enteros hasta cadenas de texto y números de punto flotante.

#### **Usos Comunes de las Listas**

- Las listas se utilizan ampliamente para almacenar y manipular datos en muchos tipos de aplicaciones.
- Por ejemplo, en una aplicación de lista de tareas, podrías usar una lista para llevar un registro de las tareas pendientes.
- En un juego, podrías utilizar una lista para almacenar la posición de los enemigos en un nivel.



## Conjuntos en Python

#### ¿Qué son los Conjuntos?

- Los conjuntos son colecciones no ordenadas de elementos únicos.
- A diferencia de las listas, los elementos en un conjunto no tienen un orden específico y no se pueden acceder por índice.
- Los conjuntos se definen utilizando llaves `{}` o la función `set()`.

## Características Clave de los Conjuntos

- Elementos Únicos: Los conjuntos no permiten duplicados. Cada elemento en un conjunto es único.
- No Ordenados: Los elementos en un conjunto no tienen un orden específico. No puedes asumir un orden particular.
- Eficiencia en Búsquedas: Los conjuntos son eficientes para verificar si un elemento está presente en la colección.



## Ejemplo de Conjunto

```
python

mi_conjunto = {1, 2, 3, 4, 5}
```

Hemos creado un conjunto llamado 'mi\_conjunto' que contiene números enteros.

Si intentáramos agregar un elemento duplicado, como 2, se ignoraría

## Usos Comunes de los Conjuntos

- Los conjuntos se utilizan cuando necesitas almacenar elementos únicos y no te importa el orden.
- Algunos ejemplos de uso incluyen:
  - Mantener una lista de elementos únicos, como direcciones de correo electrónico.
  - Verificar la existencia de elementos en una colección sin preocuparte por el orden.
  - Realizar operaciones matemáticas de conjuntos como unión, intersección y diferencia.



## Diccionarios en Python

- ¿Qué son los Diccionarios?
- Los diccionarios son colecciones no ordenadas de datos que se almacenan en pares clave-valor.
- Cada elemento en un diccionario tiene una clave única que se utiliza para acceder a su valor correspondiente.
- Se definen utilizando llaves `{}`.

## Características Clave de los Diccionarios

- Asociación de Datos: Los diccionarios permiten asociar datos en pares clave-valor, lo que facilita la recuperación eficiente de valores a través de sus claves.
- Flexibilidad de Tipos de Datos: Los valores en un diccionario pueden ser de cualquier tipo de dato, incluyendo números, cadenas de texto, listas, u otros diccionarios.
- Acceso Rápido: Los diccionarios son eficientes para buscar y acceder a valores utilizando sus claves únicas.



## Ejemplo de Diccionario

python

mi\_diccionario = {"nombre": "Juan", "edad": 30, "ciudad": "Ejemploville"}

 Hemos creado un diccionario llamado `mi\_diccionario` con tres pares clave-valor que representan información personal.

# Usos Comunes de los Diccionarios

- Los diccionarios son esenciales en la programación para:
  - Representar información estructurada, como configuraciones o datos de usuario.
  - Realizar búsquedas eficientes de valores utilizando claves.
  - Almacenar datos en una forma que sea fácilmente comprensible y manipulable.

