



Desarrollo de Aplicaciones con Objetos

Plantel Docente

- ✓ Diego Javier Serrano. dserrano@frc.utn.edu.ar
- ✓ Martin Gustavo Cassatti. mcasatti@frc.utn.edu.ar
- ✓ Fernando Mario Sanabria. fsanabria@frc.utn.edu.ar
- ✓ Oscar Ernesto Botta. obotta@frc.utn.edu.ar



Contenidos Semana 2

Temas

- ✓ Uso de archivos plano
- ✓ Lectura y manipulacion de archivos txt y csv
- ✓ Lectura y manipulacion de archivos json
- ✓ Conjuntos
- ✓ Diccionarios



Material de la semana

✓ Aula Virtual
DAO 2025

✓ Repositorio
[Github](#)

✓ Material Conceptual

https://uv.frc.utn.edu.ar/pluginfile.php/918257/mod_resource/content/4/unidad1.pdf

✓ Tutorial Visual Studio Code

https://uv.frc.utn.edu.ar/pluginfile.php/918408/mod_resource/content/2/vsc.pdf



Material de la semana (II)

✓ Videos – Tutoriales

✓ Playlist <https://www.youtube.com/@DAO-UTN>

✓ Semana 3:

<https://www.youtube.com/playlist?list=PLCYPq4p46CyTmFzuL4xIj8QWSCqdYuPCP>

- Conjuntos

○ <https://www.youtube.com/watch?v=He5jsaatc4Y>

- Operaciones de conjuntos

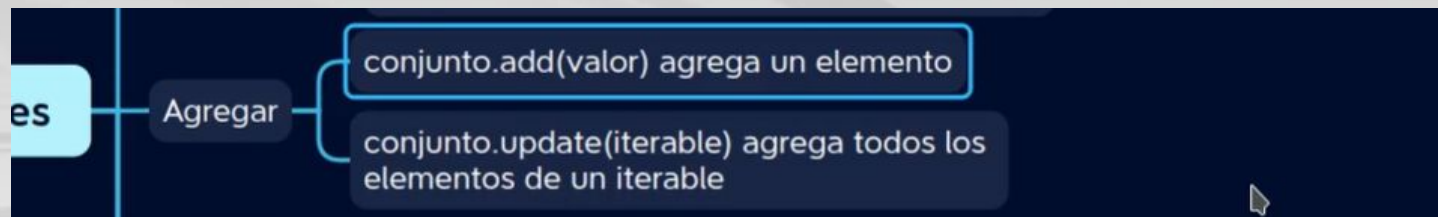
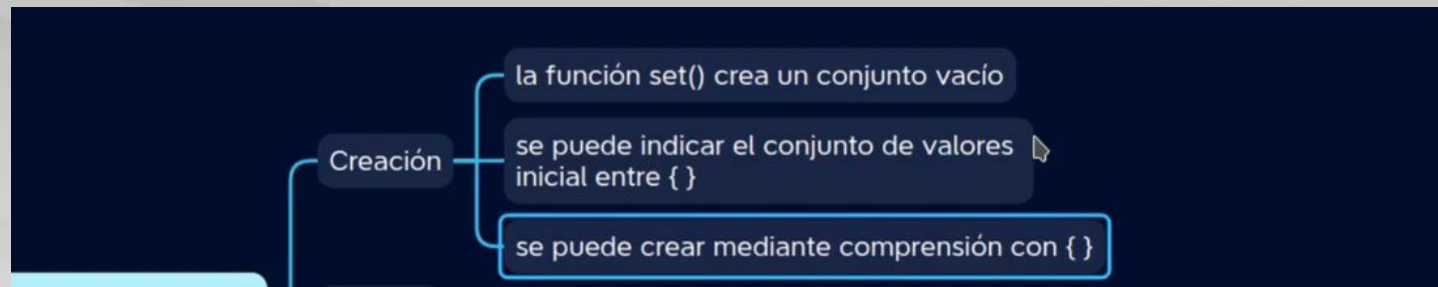
○ <https://www.youtube.com/watch?v=GNJX9XaDzm4>

- Diccionarios

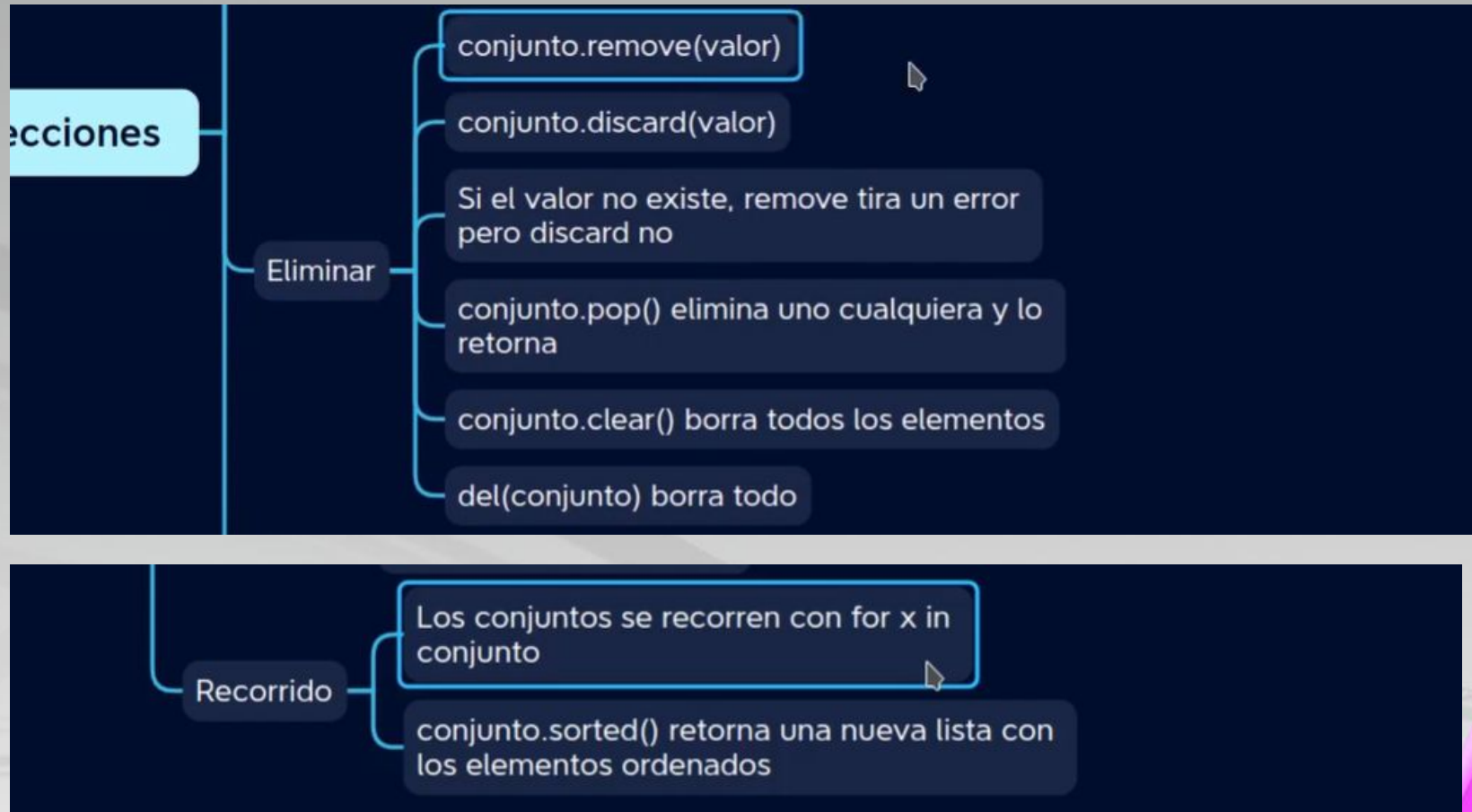
○ <https://www.youtube.com/watch?v=5QVFtNlX9iI>



Conjuntos - Metodos



Conjuntos - Metodos (II)



Conjuntos - Operaciones

Pertenencia

El operador `in` devuelve un boolean indicando si un elemento pertenece o no a un conjunto

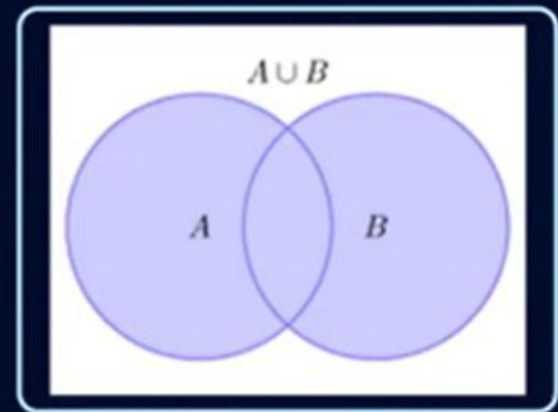
`if valor in conjunto:`

Unión

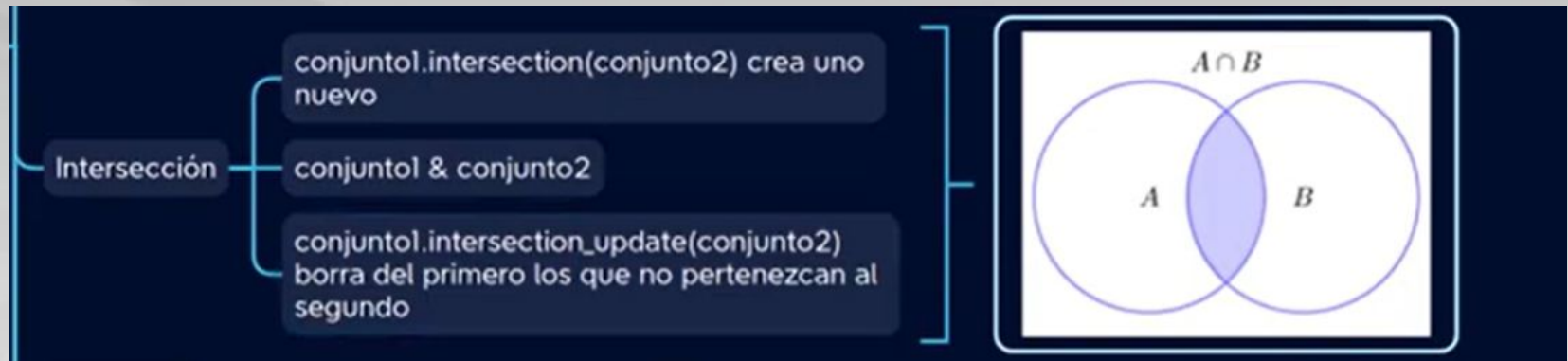
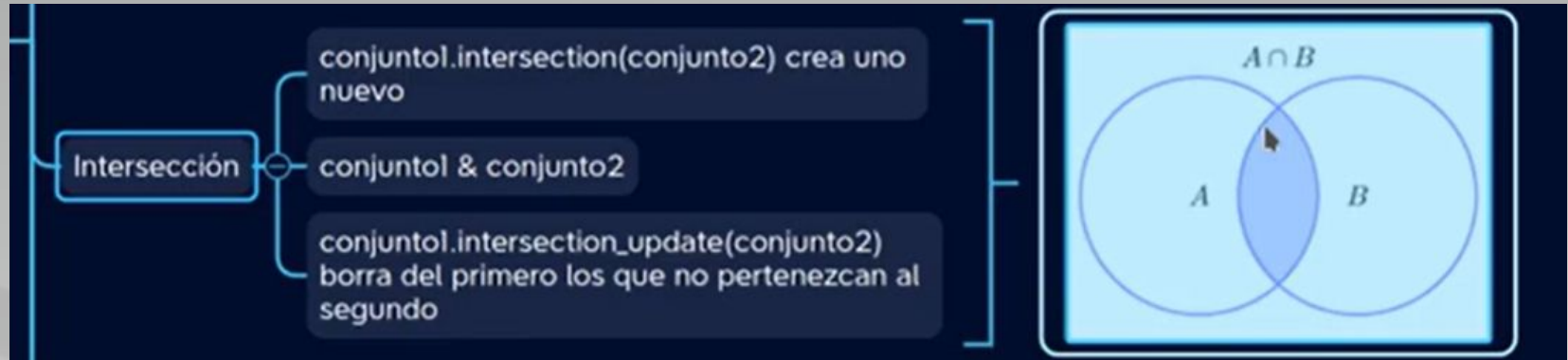
`conjunto1.union(conjunto2)` crea uno nuevo

`conjunto1 | conjunto2`

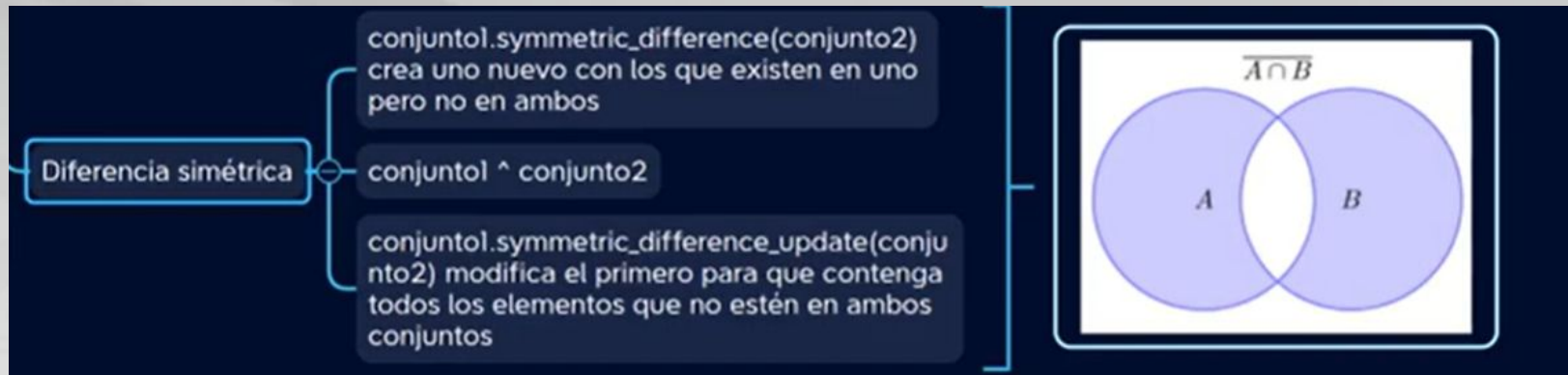
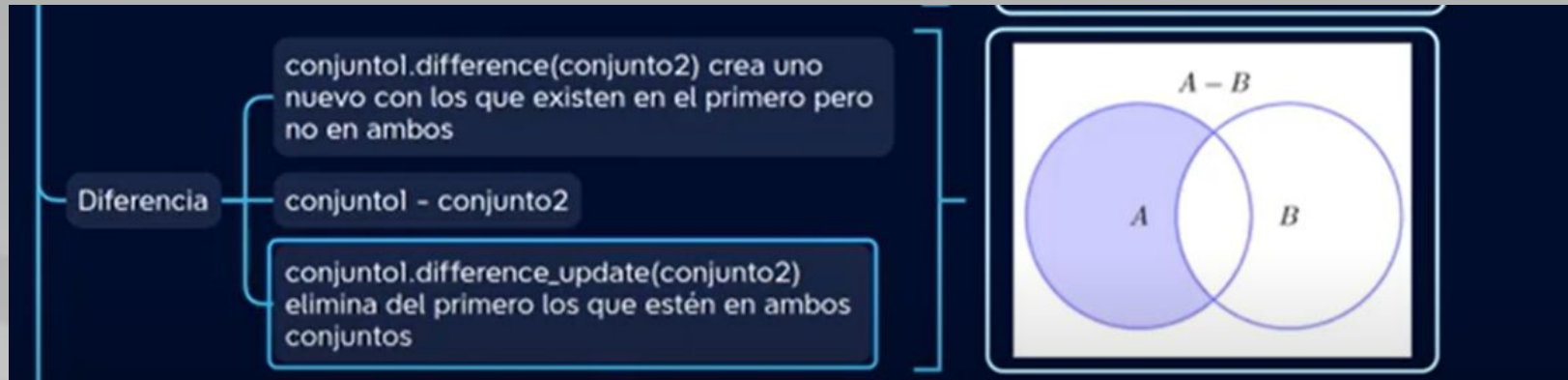
`conjunto1.update(conjunto2)` agrega al primero



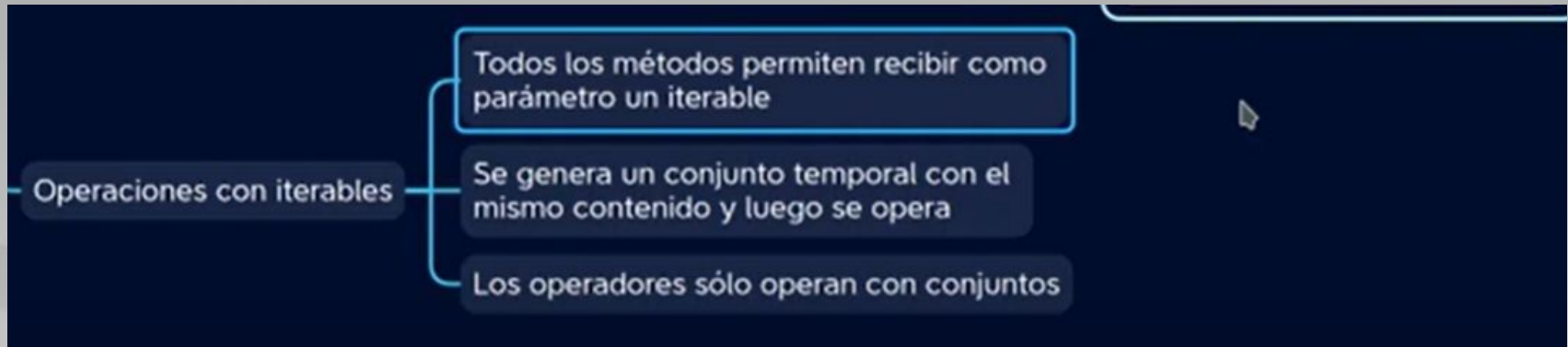
Conjuntos - Operaciones (II)



Conjuntos - Operaciones (III)



Conjuntos - Operaciones (IIII)



Resumen Colecciones avanzadas: Conjuntos

Los conjuntos son una estructura de datos proporcionada por el lenguaje que permite almacenar y manipular colecciones de elementos.

A diferencia de las listas o las tuplas, los conjuntos tienen características distintivas que los hacen únicos:

- Elementos únicos
- No ordenados
- Mutables (pueden cambiar luego de creados)
- Admiten operaciones de conjuntos:
 - `union()`
 - `intersection()`
 - `difference()`
- Heterogéneos (elementos de distintos tipos)

Resumen: Colecciones avanzadas: Conjuntos

- Los conjuntos se crean con la función `set()` o simplemente colocando llaves `{}`
- La sintaxis `{}`, puede ser utilizada con o sin elementos.
- Hay que tener en cuenta que si se crea un conjunto con elementos duplicados, los mismos se ignoran:
 - `a = {1,2,5,2,7,3,1}`
 - `print(a)` # 1,2,5,7,3

Diccionarios

Concepto:

Los diccionarios son una estructura de datos versátil que permite almacenar pares de datos conocidos como <clave= y <valor=

No permiten duplicados y son de muy rápido acceso. Velocidad de búsqueda constante.

No poseen índices como las listas o arreglos.

Simulan arreglos con índices que no son números

Métodos

- Creación. dict(). Y asignación de pares con llaves con la clave y valor separados con el signo “:” como indicador.



Diccionarios. Métodos

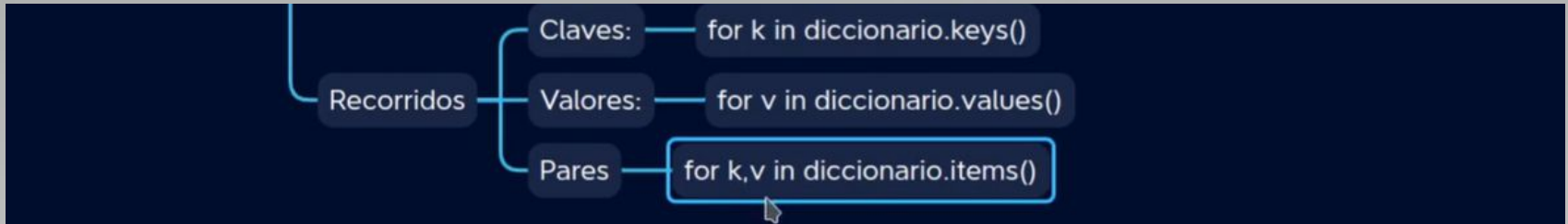
Métodos: (cont)

- Creación. dict(). Y asignación de pares con llaves con la clave y valor separados con el signo ":" como indicador.
- Búsqueda. Retorna el valor que posee la clave que se pasa entre corchetes [].Ej: diccionario["ene"]
- Inserción y Modificación. diccionario["ene"]= <valor>
- Borrado. del diccionario[<clave><default>] y diccionario.pop[<clave>] que retorna el valor.
- Recorrido. Retorna claves (keys), valores (values) y pares (items).

```
diccionario = {1: "Lunes", 2: "Martes", 3: "Miércoles"}  
  
for numero, nombre in diccionario.items():  
    print(f"El día {numero} se llama {nombre}")
```



Diccionarios. Operaciones



Operaciones:

- Unión
- Intersección
- Diferencia
- Diferencia Simétrica
- Todos se pueden aplicar a los resultados de keys, values, items.



Diccionarios. Resumen

