

ConjuntoADT

```
class ConjuntoADT:
    def __init__(self):
        self.elementos = []

    def __str__(self):
        return '{' + ', '.join(str(e) for e in self.elementos) + '}'

    def longitud(self):
        return len(self.elementos)

    def contiene(self, elemento):
        return elemento in self.elementos

    def agregar(self, elemento):
        if elemento not in self.elementos:
            self.elementos.append(elemento)

    def eliminar(self, elemento):
        if elemento in self.elementos:
            self.elementos.remove(elemento)

    def equals(self, otro_conjunto):
        if self.tamaño() != otro_conjunto.tamaño():
            return False
        for elemento in self.elementos:
            if elemento not in otro_conjunto.elementos:
                return False
        return True

    def esSubconjunto(self, otro_conjunto):
        for elemento in self.elementos:
            if elemento not in otro_conjunto.elementos:
                return False
        return True

    def union(self, otro_conjunto):
        nuevo_conjunto = ConjuntoADT()
        nuevo_conjunto.elementos = self.elementos.copy()
        for elemento in otro_conjunto.elementos:
            if elemento not in nuevo_conjunto.elementos:
                nuevo_conjunto.elementos.append(elemento)
        return nuevo_conjunto
```

ConjuntoADT

```
def interseccion(self, otro_conjunto):
    nuevo_conjunto = ConjuntoADT()
    for elemento in self.elementos:
        if elemento in otro_conjunto.elementos:
            nuevo_conjunto.agregar(elemento)
    return nuevo_conjunto

def diferencia(self, otro_conjunto):
    nuevo_conjunto = ConjuntoADT()
    for elemento in self.elementos:
        if elemento not in otro_conjunto.elementos:
            nuevo_conjunto.agregar(elemento)
    return nuevo_conjunto

def es_vacio(self):
    return len(self.elementos) == 0

#Implementación

#Creación del conjunto uno (felinos)
conjunto1 = ConjuntoADT()
conjunto1.agregar("gato")
conjunto1.agregar("león")
conjunto1.agregar("puma")

#Creación del conjunto dos (mascotas)
conjunto2 = ConjuntoADT()
conjunto2.agregar("gato")
conjunto2.agregar("perro")

print("---Creación de los conjuntos ---")
print("Conjunto 1:", conjunto1)
print("Conjunto 2:", conjunto2)

#Agregar a conjunto
conjunto1.agregar("caracal")
conjunto2.agregar("perico")

print("---Agregación a los dos conjuntos---")
print("Conjunto 1:", conjunto1)
print("Conjunto 2:", conjunto2)
```

ConjuntoADT

```
#Eliminar elemento de conjunto
conjunto1.eliminar("puma")

print("---Eliminación de elementos---")
print("Conjunto 1:", conjunto1)
print("Conjunto 2:", conjunto2)

#Pertenencia de un elemento a un conjunto
print("---Pertenencia a un conjunto---")
elemento_pertenece = "tortuga"
# Verificar si el elemento está en el conjunto
if conjunto1.contiene(elemento_pertenece):
    print(f"El elemento {elemento_pertenece} pertenece al conjunto")
else:
    print(f"El elemento {elemento_pertenece} NO pertenece al conjunto")

elemento_pertenece = "perico"
# Verificar si el elemento está en el conjunto
if conjunto 2.contiene(elemento pertenece):
    print(f"El elemento {elemento pertenece} pertenece al conjunto")
else:
    print(f"El elemento {elemento pertenece} NO pertenece al conjunto")

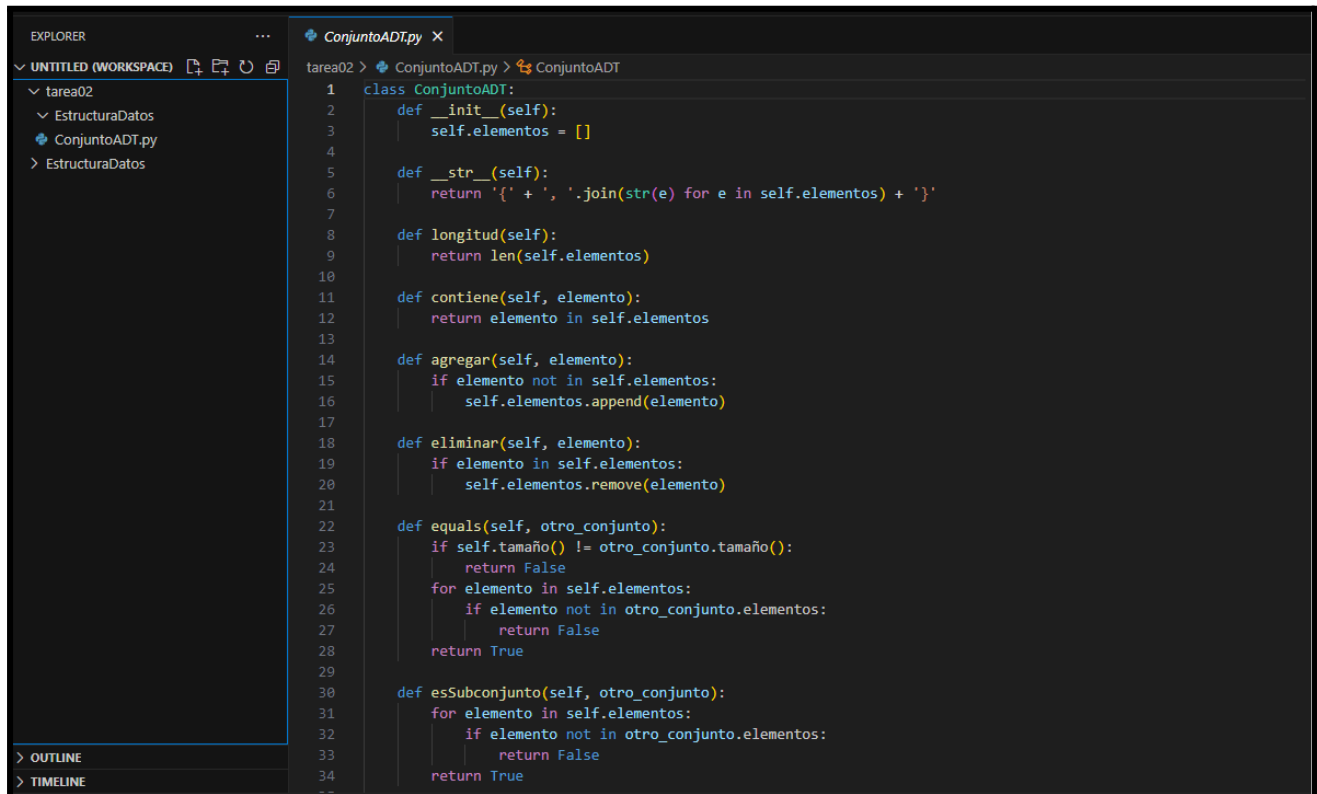
#Unión de dos conjuntos
union_conjuntos = conjunto1.union(conjunto2)
print("---Unión:", union_conjuntos)

#Intersección entre conjuntos
interseccion_conjuntos = conjunto1.interseccion(conjunto2)
print("---Intersección:", interseccion_conjuntos)

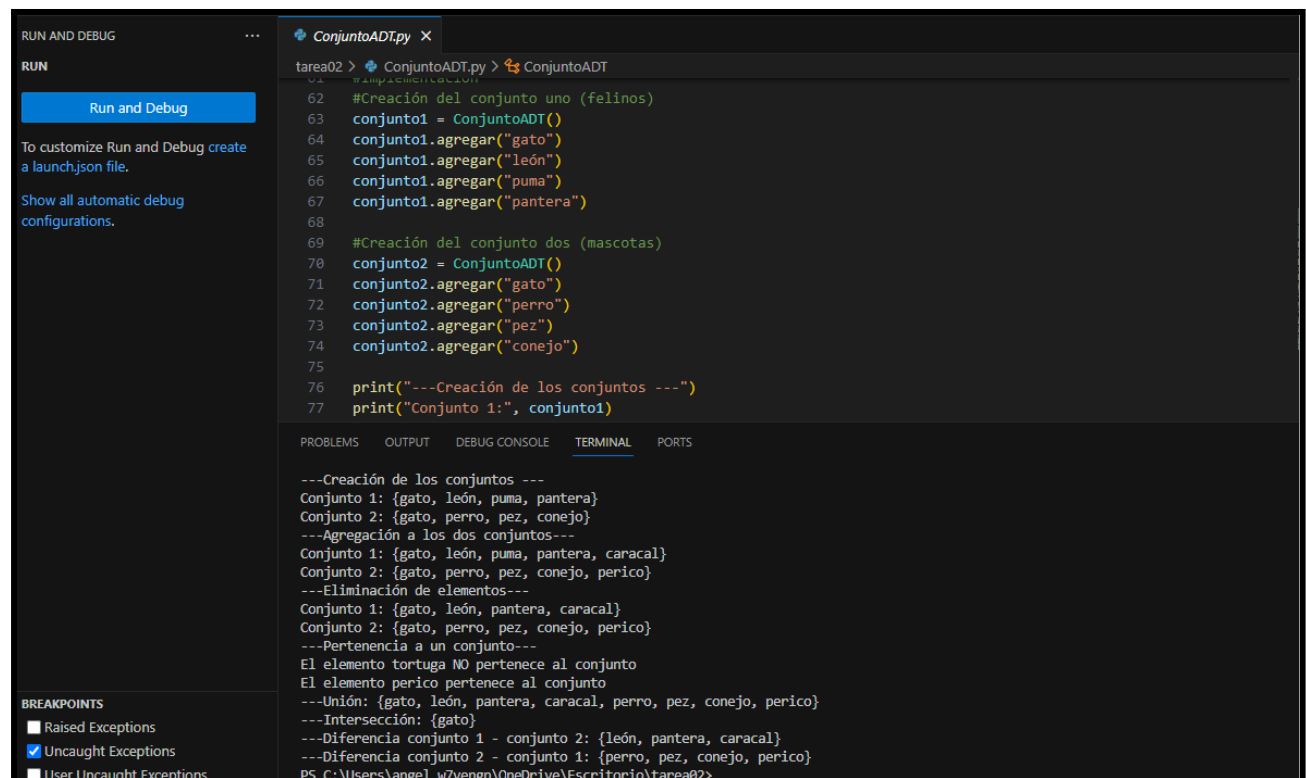
#Diferencia entre conjuntos
diferencia_conjuntos = conjunto1.diferencia(conjunto2)
print("---Diferencia:", diferencia_conjuntos)
```

ConjuntoADT

Capturas de pantalla



```
1 class ConjuntoADT:
2     def __init__(self):
3         self.elementos = []
4
5     def __str__(self):
6         return '{' + ', '.join(str(e) for e in self.elementos) + '}'
7
8     def longitud(self):
9         return len(self.elementos)
10
11    def contiene(self, elemento):
12        return elemento in self.elementos
13
14    def agregar(self, elemento):
15        if elemento not in self.elementos:
16            self.elementos.append(elemento)
17
18    def eliminar(self, elemento):
19        if elemento in self.elementos:
20            self.elementos.remove(elemento)
21
22    def equals(self, otro_conjunto):
23        if self.tamaño() != otro_conjunto.tamaño():
24            return False
25        for elemento in self.elementos:
26            if elemento not in otro_conjunto.elementos:
27                return False
28        return True
29
30    def esSubconjunto(self, otro_conjunto):
31        for elemento in self.elementos:
32            if elemento not in otro_conjunto.elementos:
33                return False
34        return True
```



```
62 #Creación del conjunto uno (felinos)
63 conjunto1 = ConjuntoADT()
64 conjunto1.agregar("gato")
65 conjunto1.agregar("león")
66 conjunto1.agregar("puma")
67 conjunto1.agregar("pantera")
68
69 #Creación del conjunto dos (mascotas)
70 conjunto2 = ConjuntoADT()
71 conjunto2.agregar("gato")
72 conjunto2.agregar("perro")
73 conjunto2.agregar("pez")
74 conjunto2.agregar("conejo")
75
76 print("---Creación de los conjuntos ---")
77 print("Conjunto 1:", conjunto1)
```

---Creación de los conjuntos ---
Conjunto 1: {gato, león, puma, pantera}
Conjunto 2: {gato, perro, pez, conejo}
---Agregación a los dos conjuntos---
Conjunto 1: {gato, león, puma, pantera, caracal}
Conjunto 2: {gato, perro, pez, conejo, perico}
---Eliminación de elementos---
Conjunto 1: {gato, león, pantera, caracal}
Conjunto 2: {gato, perro, pez, conejo, perico}
---Pertenencia a un conjunto---
El elemento tortuga NO pertenece al conjunto
El elemento perico pertenece al conjunto
---Unión: {gato, león, pantera, caracal, perro, pez, conejo, perico}
---Intersección: {gato}
---Diferencia conjunto 1 - conjunto 2: {león, pantera, caracal}
---Diferencia conjunto 2 - conjunto 1: {perro, pez, conejo, perico}
PS C:\Users\angel w\venen\OneDrive\Escritorio\tarea02>