

Documentación de Prueba Técnica 4

Prueba

Se han determinado las siguientes tuplas, las cuales representan una relación entre elementos

jurídicos:

```
relaciones = [  
    ('Principio de Contradicción', 'Sentencia 338/2018'),  
    ('Juez de Amparo', 'Tribunal Colegiado'),  
    ('Auto de Vinculación a Proceso', 'Juez de Control'),  
    ('Organo Jurisdiccional', 'Juez de Control'),  
    ('Juzgado de Control', 'Sentencia 338/2018'),  
    ('Juzgado de Distrito', 'Sentencia 338/2018'),  
    ('Norma Convencional que contempla justa indemnización',  
     'Sentencia 423/2019'),  
    ('Autoridad Jurisdiccional', 'Suprema Corte de Justicia de la Nación'),  
    ('Organo Jurisdiccional', 'Juez de Amparo'),  
    ('Norma Convencional que contempla justa indemnización',  
     'Artículo 63.1 Convención Americana sobre Derechos Humanos'),  
    ('Juez de Amparo', 'Juzgado de Distrito'),  
    ('Tribunal Colegiado', 'Sentencia 338/2018'),  
    ('Norma Convencional que contempla justa indemnización',  
     'Norma Convencional'),  
    ('Prisión Preventiva Oficiosa', 'Organo Jurisdiccional'),  
    ('Autoridad Jurisdiccional', 'Organo Jurisdiccional')  
]
```

Realiza una función que reciba como entrada dos de los elementos jurídicos anteriormente mencionados (elementoJuridicoInicial, elementoJuridicoFinal) y como salida de una lista de elementos que muestre la ruta de conexión de los dos elementos dados. Ejemplo:

input(elemento1, elemento5) --> output: [elemento1, elemento3, elemento5]

Solución

Código

Se importa el módulo random para generar selecciones aleatorias.

```
import random
```

Se define la lista de relaciones.

```
relaciones = [  
    ('Principio de Contradicción', 'Sentencia 338/2018'),  
    ('Juez de Amparo', 'Tribunal Colegiado'),  
    ('Auto de Vinculación a Proceso', 'Juez de Control'),  
    ('Organo Jurisdiccional', 'Juez de Control'),  
    ('Juzgado de Control', 'Sentencia 338/2018'),  
    ('Juzgado de Distrito', 'Sentencia 338/2018'),  
    ('Norma Convencional que contempla justa indemnizacion', 'Sentencia 423/2019'),  
    ('Autoridad Jurisdiccional', 'Suprema Corte de Justicia de la Nación'),  
    ('Organo Jurisdiccional', 'Juez de Amparo'),  
    ('Norma Convencional que contempla justa indemnizacion', 'Artículo 63.1 Convención Americana  
sobre Derechos Humanos'),  
    ('Juez de Amparo', 'Juzgado de Distrito'),  
    ('Tribunal Colegiado', 'Sentencia 338/2018'),  
    ('Norma Convencional que contempla justa indemnizacion', 'Norma Convencional'),  
    ('Prisión Preventiva Oficiosa', 'Órgano Jurisdiccional'),  
    ('Autoridad Jurisdiccional', 'Órgano Jurisdiccional')  
]
```

Se seleccionan aleatoriamente dos elementos únicos de la lista de relaciones.

```
elemento1, elemento2 = random.sample(set([relacion[0] for relacion in relaciones]), 2)
```

Se crea una lista llamada "elementos_relacionados" para almacenar los elementos relacionados con los elementos seleccionados.

```
elementos_relacionados = []
```

Se recorre la lista de relaciones y se agregan los elementos relacionados a la lista "elementos_relacionados".

```
for relacion in relaciones:
```

```
    if relacion[0] == elemento1 or relacion[0] == elemento2:
```

```
        elementos_relacionados.append(relacion[1])
```

```
    if relacion[1] == elemento1 or relacion[1] == elemento2:
```

```
        elementos_relacionados.append(relacion[0])
```

Se selecciona aleatoriamente uno de los elementos relacionados como el tercer elemento de la relación.

```
elemento3 = random.choice(elementos_relacionados)
```

Se imprime en pantalla los elementos de entrada (elemento1 y elemento2) y los elementos de salida (elemento1, elemento2 y elemento3).

```
print("los elementos input ", "(" , elemento1 , ")", "(" , elemento2 , ")")
```

```
print("los elementos output ", "(" , elemento1 , ")", "(" , elemento2 , ")", "(" , elemento3 , ")")
```

El código genera una salida que muestra los elementos de entrada y los elementos de salida basados en las relaciones aleatorias selección