Práctica 4:

LISTAS

OBJETIVO

- Uso de listas en Prolog.
- Recursividad.
- Uso de la aritmética de Prolog.

PLANTEAMIENTO DEL PROBLEMA

Implementar los siguientes predicados sobre listas:

1. suma_L/2: Predicado que suma todos los elementos de una lista de números.

Ejemplo de utilización:

```
suma_L([1,3,2], S).                ->    S = 6
```

2. miembro/2: Predicado que tiene como parámetros un elemento y una lista y es cierto si el elemento está en la lista y falso en caso contrario.

Ejemplo de utilización:

```
miembro(3, [1,3,2]).               ->    yes
miembro(4, [1,3,2]).               ->    no
```

3. borrar/3: Predicado que a partir de un elemento y una lista, devuelve la lista sin una de las apariciones del elemento. Si el elemento no aparece devuelve la lista original.

Ejemplo de utilización:

```
borrar(3, [1,3,2,3], L).           ->    L = [1,2,3]
```

4. unir/3: Predicado que a partir de dos listas devuelve una tercera que es la concatenación.

Ejemplo de utilización:

```
unir([1,2], [1,3], L).             ->    L = [1,2,1,3]
```

Estos predicados se deben implementar utilizando recursividad, y como en cualquier algoritmo recursivo, existirá un caso directo y un caso recursivo. Cada uno de estos casos debe corresponder con una cláusula de Prolog.

5. inserta_ult/3: Predicado que inserta un elemento al final de una lista.

Ejemplo de utilización:

```
inserta_ult(3,[2,4,5],L).          ->    L= [2,4,5,3]
```

6. elimina_repe/2: Predicado que elimina los elementos repetidos de una lista.

Ejemplo de utilización:

```
elimina_repe([2,1,3,2,1,2], L).    ->    L = [2,1,3]
```