**Práctica 3: Programación lógica**

**Ejercicio 1**

**Declarativa:**

1. Dada la lista que comienza por el elemento X, la sublista entre los índices 1 y 1 es [X].

2. Si la sublista de Xs entre el índice 1 y el índice K-1, K>1, es Ys, entonces la sublista de la lista [X|Xs] entre los índices 1 y K es [X|Ys].

3. Si la sublista de Xs entre los índices I-1 y K-1, I>1, es Ys, entonces la sublista entre los índices I y K de la lista [\_|Xs] es Ys.

**Procedural (slice([1, 2, 3 ,4],2,3,L2)):**

1. El primer caso no se ejecuta porque 2!=1 y 3!=1.

2. El segundo caso no se ejecuta porque 2!=1.

3. El tercero sí que se ejecuta. Se realiza la siguiente unificación: {Xs:=[2,3,4], I:=2, K:=3}. Por tanto I1=I-1=1, K1=K-1=2, obteniendo la nueva consulta “3>1 and slice([2,3,4],1,2,L2)”. La primera es verdadera.

4. El primer caso no se ejecuta porque 2!=1.

5. El segundo caso sí que se ejecuta. Se realiza la siguiente unificación: {X:=2, Xs:=[3,4], K:=2}. L2 toma el valor [2|Ys]. Por tanto K1=K-1=1, obteniendo la nueva consulta “2>1 and slice([3,4],1,1,Ys)”. La primera es verdadera.

6. El primer caso sí que se ejecuta. Se realiza la siguiente unificación: {X:=3}. Ys toma el valor [3].

7. L2 queda por tanto [2|Ys] = [2,3].

**Ejercicio 9.5**

Para hacer el estudio de cómo afecta el valor de K a la tasa de acierto del algoritmo “leave one out” hemos creado un predicado lista\_tasa\_aciertos/30, que muestra en 30 listas los pares [K, tasa] con K desde 1 hasta 149. Los resultados son los siguientes:



La tasa de acierto se maximiza en torno a K=20 con un valor del 98% y a partir de ahí decrece hasta un valor aproximado del 85% en torno a K=95 y a partir de ese valor decae muy rápido hasta alcanzar el 0% en los valores finales.