Práctica 1 de Lenguajes – Paradigma Lógico

Ejercicios

**NOTAS:**

* **Los siguientes ejercicios pretenden que el programador se enfoque en el algoritmo que resuelve el problema y no en la interacción con el usuario para obtener datos, así que se pueden trabajar con datos quemados y haciendo pruebas de funciones/métodos controladas por el programador en el main de los programas, si se requiere.**
* **Los ejercicios deben ser implementados por el/la estudiante y no utilizar otras funciones que existan que los realicen ya en Prolog (si es que existe)**
* **Todos los ejercicios deben subirse al repositorio**

1. Implemente un predicado que, a partir de una lista de cadenas de parámetro, filtre aquellas que contengan una subcadena que el usuario indique en otro argumento. Ej

sub\_cadenas(“la”, [“la casa, “el perro”, “pintando la cerca”],Filtradas).

True

Filtradas = [“la casa, “pintando la cerca”]

1. Construya una función que se llame sub\_conjunto. Esta función recibe dos listas y debe retornar True cuando el primer argumento es subconjunto completo del segundo y #f en caso contrario. Por ejemplo:

sub\_conjunto([],[1,2,3,4,5]).

True

sub\_conjunto([1,2,3],[1,2,3,4,5]).

True

sub\_conjunto([1,2,6],[1,2,3,4,5]).

False

1. Implemente la función aplanar. Esta función recibe una lista con múltiples listas anidadas como elementos y devuelve una lista con los mismos elementos de manera lineal (sin listas). Ej:

aplanar([1,2,[3,[4,5],[6,7]]],X).

True

X=[1,2,3,4,5,6,7].

1. Modifique el predicado para averiguar las rutas en un grafo para que calcule la ruta más corta de todas en función de pesos que deben agregarse al grafo.