|  |  |  |  |
| --- | --- | --- | --- |
| **Alumno(s):** | Gerardo Pantoja Méndez | **Fecha:** | 21/11/17 |

|  |
| --- |
| OBJETIVO |
| *Resolver los problemas planteados usando el lenguaje de programación lógica ProLog. Para adquirir más conocimientos sobre este lenguaje.* |

|  |
| --- |
| MARCO TEÓRICO |
| *Prolog es un lenguaje de programación lógico e interpretado usado habitualmente en el campo de la Inteligencia artificial.*  *En Prolog, las instrucciones se ejecutan normalmente en orden secuencial, es decir, una a continuación de otra, en el mismo orden en que están escritas, que sólo varía cuando se alcanza una instrucción de control (un bucle, una instrucción condicional o una transferencia).*  *En Prolog no existen instrucciones de control. Su ejecución se basa en dos conceptos: la unificación y el backtracking.* |

|  |
| --- |
| DESARROLLO |
| ***Problema 1.*** *Monty Python & the Holy Grail (1975)*  *Este problema trata de correlacionar los patos y las brujas, diciendo que ambos flotan, que las brujas flotan porque son de madera.*  ***witch(X) :- burns(X),female(X).***  ***burns(X) :- wooden(X).***  ***wooden(X) :- floats(X).***  ***floats(X) :- sameweight(duck, X).***    ***female(girl).***  ***sameweight(duck,girl).***    ***Problema 2: Canción estúpida***  *Aquí solo debemos enviar un parámetro que configuramos dentro del programa, con esto el Prolog, mostrará la letra de una canción estúpida.*  *Convirtiendose en un ciclo, si ponemos un numero mayor a 1, hasta que lleguemos a 1.*  ***cancionestupida(0):-nl,write('Gomo ya no gueda shevvezza, -hic- be boy a doddmig...').***    ***cancionestupida(N):-N>1,nl,write(N),write(' botellas de cerveza en el suelo'),nl,***  ***write(N),write(' botellas de cerveza'),nl,***  ***write('Cojo una y me la bebo'),nl,***  ***A is N-1, cancionestupida(A).***    ***cancionestupida(N):-N=1,nl,write(N),write(' bodellia de shegvezza en el zsduelo'),nl,***  ***write(N),write(' bodella de segbezha'),nl,***  ***write('La gojo y be la bhebo'),nl,***  ***A is N-1, cancionestupida(A).***    *Problema 3:* ***Diagnostico de enfermedades.***  *Aquí se trata de averiguar si una enfermedad tiene cierta enfermedad, en base a los síntomas que presenta.*  ***enfermo\_de(manuel,gripe).***  ***tiene\_sintoma(alicia,cansancio).***  ***sintoma\_de(fiebre,gripe).***  ***sintoma\_de(tos,gripe).***  ***sintoma\_de(cansancio,anemia).***  ***elimina(vitaminas,cansancio).***  ***elimina(aspirinas,fiebre).***  ***elimina(jarabe,tos).***  ***recetar\_a(X,Y):-enfermo\_de(Y,A),alivia(X,A).***  ***alivia(X,Y):-elimina(X,A),sintoma\_de(A,Y).***    ***enfermo\_de(X,Y):-tiene\_sintoma(X,Z),sintoma\_de(Z,Y).***  *Problema 4:* ***Planificador de gastos.***  *Aquí se trata de calcular los gastos de un viaje analizando las posibles variables que afectan a dicho costo.*  ***transporte(roma,20000).***  ***transporte(londres,25000).***  ***transporte(tunez,15000).***    ***alojamiento(hotel,roma,25000).***  ***alojamiento(hotel,londres,15000).***  ***alojamiento(hotel,tunez,10000).***  ***alojamiento(hostal,roma,15000).***  ***alojamiento(hostal,londres,10000).***  ***alojamiento(hostal,tunez,8000).***  ***alojamiento(camping,roma,10000).***  ***alojamiento(camping,londres,5000).***  ***alojamiento(camping,tunez,5000).***    ***viaje(W,X,Y,Z):-transporte(W,A),alojamiento(Y,W,C),B is C\*X,Z is A+B.***    ***Problema 5:*** *Árbol genealógico.*  *Aquí hay que crear un pequeño árbol genealógico con relaciones de padre,hijos y ancestros.*  ***juan.***  ***jose. padre(juan,jose).***  ***padre(juan,maria).***  ***padre(pedro,juan).***  ***padre(jose,luis).***  ***padre(X,Y).***  ***ancestro(X,Y):-padre(X,Y).***  ***ancestro(X,Y):-padre(X,Z),padre(Z,Y)***  ***Problema 7:*** *Horoscopo*  *Aquí debemos crear un método, para que diga si alguien pertenece a algún signo zodiacal, proporcionando el dia y mes del signo.*  ***horoscopo(aries,21,3,21,4).***  ***horoscopo(tauro,21,4,21,5).***  ***horoscopo(geminis,21,5,21,6).***  ***horoscopo(cancer,21,6,21,7).***  ***horoscopo(leo,21,7,21,8).***  ***horoscopo(virgo,21,8,21,9).***  ***horoscopo(libra,21,9,21,10).***  ***horoscopo(escorpio,21,10,21,11).***  ***horoscopo(sagitario,21,11,21,12).***  ***horoscopo(capricornio,21,12,21,1).***  ***horoscopo(acuario,21,1,21,2).***  ***horoscopo(piscis,21,2,21,3).***  ***signo(Dia,Mes,Signo) :- horoscopo(Signo,D1,M1,D2,M2),((Mes=M1,Dia>=D1); (Mes=M2,Dia=<D2)).***    ***Problema 6: Red Semántica***  *Aquí solo debemos construir una red semántica, en base a la figura que se nos proporciona en el documento y a los lineamientos de sintaxis tabien del documento.*  *.*  ***atributo(animal,puede,respirar).***  ***atributo(ave,vuela,bien).***  ***atributo(ave,tiene,plumas).***  ***atributo(ave,pone,huevos).***  ***atributo(ave,es\_un,animal).***  ***atributo(mamifero,tiene,pelo).***  ***atributo(mamifero,es\_un,animal).***  ***atributo(mamifero,da,leche).***  ***atributo(avestruz,es\_un,ave).***  ***atributo(avestruz,patas,largas).***  ***atributo(avestruz,vuela,no\_puede)***  ***atributo(albatros,es\_un,ave).***  ***atributo(albatros,vuela,muy\_bien).***  ***atributo(ballena,es\_un,mamifero).***  ***atributo(ballena,tiene,piel).***  ***atributo(ballena,vive\_en,mar).***  ***atributo(tigre,es\_un,mamifero).***  ***atributo(tigre,come,carne).***    ***Problema 9:*** *Manejo de listas*  *Aquí simplemente hacemos tareas con listas, como lo hemos venido haciendo en otros lenguajes de programación.*  *Tareas como concatenar, invertir, insertar elementos en lista o ver si una variable es una lista.*  ***miembro(X,[X|\_]).***  ***miembro(X,[\_|Y]) :- miembro(X,Y).***  ***nel([],0). nel([\_|Y],N) :- nel(Y,M), N is M+1.***  ***es\_lista([]).***  ***es\_lista([\_|\_]).***  ***concatena([],L,L).***  ***concatena([X|L1],L2,[X|L3]) :- concatena(L1,L2,L3).***  ***ultimo(X,[X]).***  ***ultimo(X,[\_|Y]) :- ultimo(X,Y).***    ***inversa([],[]).***  ***inversa([X|Y],L) :- inversa(Y,Z),concatena(Z,[X],L).***  ***borrar(X,[X|Y],Y).***  ***borrar(X,[Z|L],[Z|M]) :- borrar(X,L,M).***  ***subconjunto([X|Y],Z) :- miembro(X,Z),subconjunto(Y,Z).***  ***subconjunto([],\_).***  ***insertar(E,L,[E|L]).***  ***insertar(E,[X|Y],[X|Z]) :- insertar(E,Y,Z).***  ***permutacion([],[]).***  ***permutacion([X|Y],Z) :- permutacion(Y,L),insertar(X,L,Z).***  ***sust(\_,\_,[],[]).***  ***sust(E1,E2,[E1|L1],[E2|L2]) :- !, sust(E1,E2,L1,L2).***  ***sust(E1,E2,[Y|L1],[Y|L2]) :- sust(E1,E2,L1,L2).***  ***union([],L,L).***  ***union([X|L1],L2,L3) :- miembro(X,L2), !,union(L1,L2,L3).***  ***union([X|L1],L2,[X|L3]) :- union(L1,L2,L3).*** |

|  |
| --- |
| CONCLUSIONES |
| *En este ejercicio se pudo poner en practica lo que hemos aprendido del lenguaje ProLog, y nos pudimos dar cuenta de otras cosas como la manera en que se comportan las listas en este lenguaje, basado en expresiones, logicas.* |

|  |
| --- |
| BIBLIOGRAFÍA |
| Referencias Inteligencia en Redes de Comunicaciones Prolog. (s.f.). |