



Materia:	Programación Lógica y Funcional	
Práctica:	Ejercicio: 99 botellas de cerveza en la pared	
Alumno(s):	Luis Manuel Cárdenas Ibarra	Fecha: 01/04/2025

Objetivo

Con este trabajo se se busca realizar 3 ejercicios en Prolog, los cuales son los siguientes:

- Planificador de gastos: Calcular el costo total de un viaje considerando transporte y alojamiento.
- Horóscopo: Validar fechas de nacimiento y determinar el signo zodiacal correspondiente.
- Diagnóstico de enfermedades: Implementar un sistema experto básico para diagnosticar enfermedades y recomendar medicamentos.

Las reglas para los ejercicios ya se nos proporcionaron en el documento adjunto de la práctica.

Desarrollo

Planificador de gastos:

Primero se definieron las reglas que ya se nos había compartido.

```
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) .
```



Una vez definidas las reglas, tenemos que hacer la regla principal, la cual se encargará de obtener el costo total del viaje multiplicando el tipo de alojamiento, por los días que se durará el viaje dependiendo del lugar a donde vaya, lo podemos lograr de la siguiente forma.

costo_viaje(Lugar, TipoAlojamiento, Dias, CostoTotal) :-
 transporte(Lugar, CostoTransporte),
 alojamiento(TipoAlojamiento, Lugar, CostoPorDia),
 CostoTotal is CostoTransporte + (CostoPorDia * Dias).

Horóscopo:

Para hacer el programa horóscopo, también copiamos las reglas predefinidas.

```
/* horoscopo(Signo,DiaIni,MesIni,DiaFin,MesFin)
   <- son del signo Signo los nacidos entre el DiaIni/MesIni y el DiaFin/MesFin */
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) <- los nacidos el Dia/Mes pertenecen al signo Signo */
signo(Dia,Mes,Signo) :- horoscopo(Signo,D1,M1,D2,M2),
    ( (Mes=M1,Dia>=D1) ; (Mes=M2,Dia<=D2) ).

?- signo(8,5,tauro).

?- signo(7,8,Signo).
?- signo(7,X,Signo).
?- signo(X,7,Signo).
```



En este caso, lo que tenemos que hacer es modificar el código para validar todo el rango que abarcan los horóscopos, de la siguiente forma podemos agregar cuantos días tiene cada mes y poder saber si está dentro del rango o no.

```
dias_mes(1, 31). % enero
dias_mes(2, 28). % febrero
dias_mes(3, 31). % marzo
dias_mes(4, 30). % abril
dias_mes(5, 31). % mayo
dias_mes(6, 30). % junio
dias_mes(7, 31). % julio
dias_mes(8, 31). % agosto
dias_mes(9, 30). % septiembre
dias_mes(10, 31). % octubre
dias_mes(11, 30). % noviembre
dias_mes(12, 31). % diciembre
```

Con esto definido, podremos realizar la regla principal, la cual se va a encargar de hacer las operaciones necesarias .

```
signo(Dia, Mes, Signo) :-
    dias_mes(Mes, MaxDias),
    Dia >= 1, Dia <= MaxDias,
    horoscope(Signo, D1, M1, D2, M2),
    ( (Mes = M1, Dia >= D1) ; (Mes = M2, Dia <= D2) ).
```



Diagnóstico de enfermedades:

Igualmente, lo primero es copiar las reglas brindadas.

```
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) .
```

Con la regla “recetar_a” podemos saber que medicamento recetar a cada paciente dependiendo de lo que padezca. Con la regla “alivia” podemos saber que enfermedad se alivia con que medicamento. Por último, con la regla “enfermo_de” podemos saber que enfermedad tiene cada paciente.

```
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).
```



Resultado

Planificador de gastos:

```
?- costo_viaje(londres,hostal,5,Total).  
Total = 75000.  
  
?- costo_viaje(tunez,camping,3,Total).  
Total = 30000.  
  
?- costo_viaje(roma,hotel,4,Total).  
Total = 120000.
```

Horóscopos:

```
?- signo(4,11,escorpio).  
true.  
  
?- signo(4,4,sagitario).  
false.  
  
?- signo(4,4,aries).  
true.  
  
?- signo(26,3,piscis).  
false.  
  
?- signo(26,3,aries).  
true.  
  
?- ■
```

Diagnóstico de enfermedad:

```
?- recetar_a(Medicamento,manuel).  
Medicamento = aspirinas ;  
Medicamento = jarabe ;  
false.  
  
?- alivia(aspirinas,Enfermedad).  
Enfermedad = gripe.  
  
?- enfermo_de(alicia,Enfermedad).  
Enfermedad = anemia.
```



Conclusiones

Los ejercicios resolvieron problemas diversos usando **lógica de predicados**, demostrando la versatilidad de Prolog para:

- Cálculos estructurados (viajes).
- Validación de datos (fechas y horóscopos).
- Razonamiento automático (diagnóstico médico).

Este trabajo evidenció cómo Prolog permite implementar soluciones claras y eficientes para problemas que requieren manejo de relaciones lógicas y validación de restricciones, destacando su potencial en aplicaciones de inteligencia artificial y sistemas expertos.