



PROYECTO 1

CREACIÓN DE SISTEMA EXPERTO EN PROLOG

ARTIFICIAL INTELLIGENCE

Yaiza Pineda
201610673



Objetivos

General:

Ampliar e implementar los conocimientos adquiridos en el laboratorio sobre la programación lógica generando un sistema experto

Específicos:

Aprender los conceptos generales para aprender a programar en el lenguaje Prolog

Realizar un programa que sea capaz de sugerir hoteles basándose en los datos ingresados y el análisis realizado.

Generar 7 reportes con base a los datos y el reporte seleccionado.

Comprender la importancia de resolver problemas usando el cálculo de predicados.





Detalle de Recursos Utilizados

Prolog

Un programa Prolog se compone de un conjunto de hechos. Herramienta que se utilizo para realizar el desarrollo del sistema experto para la gestión de opción de viaje.



Sublime

Es un editor de texto para escribir código. Se utilizó sublime text para generar el código implementado, esto para tener una mayor facilidad de manipular las líneas de código.



Access

Es un gestor de datos que utiliza los conceptos de bases de datos relacionales y pueden mane jarse por medio de consultas e informes. Utilizado para obtener los datos de la base de datos de cada una de las tablas proporcionadas.



Excel

Es un programa para el análisis de datos. Se basa en una hoja de cálculo. Excel permitió manipular los datos obtenido de la base de datos en access para poder generar fácilmente los hechos.



Recursos del PC:

Ram : 4 GB

SO: Windows 10 de 64bits

Procesador: Intel Core i5

SWI-Prolog version (threaded, 64 bits, version 8.4.2)

CPU: 0%

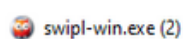
Memoria: 17.1 MB

Disco: 0 MB/s

Red: 0 Mbps

Consumo de energía: Muy baja

Tendencia de consumo de energía: Muy baja



0%

4,8 MB

0 MB/s

0 Mbps

Muy baja

Muy baja



Explicación del código de la solución

Especificaciones Generales

- write -> utilizado para la salida en consola
- read -> utilizado para leer la entrada en teclado
- comentarios inician con la letra %
- Para realizar expresiones aritméticas utilizamos el is
- Las variables no son globales y son declaradas con inicial mayúscula.
- Para definir reglas se utiliza, cabeza : - cuerpo
- Para realizar condicionales, variable -> hechos con conjunciones

Desarrollo:

Empezamos declarando los hechos en minúscula, entre paréntesis añadimos la relación y se debe finalizar con un .

```
% --- Hechos de Departamento ---  
departamento(1,peten,8,español,'tropical').
```

Para el menu necesitamos definir una regla, dentro de la cual utilizaremos write para mostrar en pantalla las opciones y al final un read para capturar la respuesta del usuario

```
% --- Menu -----  
menu:- repeat,nl,  
    write('----- Bienvenidos -----'),nl,  
    write(' ¡Te brindamos tu mejor opción para que disfrutes tus vacaciones! '),nl,  
    write('1. Presupuesto '),nl,  
    write('2. Idioma '),nl,  
    write('3. Cantidad de estrellas '),nl,  
    write('4. Clima '),nl,  
    write('5. Reportes '),nl,  
    read(OPCION),nl,
```

Creamos una regla con un parámetro que nos redireccionara dependiendo la opcion elegida por el usuario

```
ejecutar(OPCION):-(  
    OPCION == 1 -> criPresupuesto();  
    OPCION == 2 -> criIdioma();  
    OPCION == 3 -> criEstrellas();  
    OPCION == 4 -> criClima();  
    OPCION == 5 -> criReportes();
```



Explicación del código de la solución

Al momento de seleccionar un camino se inicia con una serie de interrogantes para obtener los requerimientos que el usuario desea establecer para su viaje.

Posteriormente usamos una regla para filtrar las respuestas, enfocándonos en si se transportara en vehículo y el tipo de habitación elegida.

```
analisis(Presupuesto,Diash,Thabitacion,Distancia,Vehiculo,Tcomida,Clima):-(  
(Vehiculo == 's',Thabitacion = 1)->elegircvhsimple(Presupuesto,Diash,Distancia,Tcomida,Clima);  
(Vehiculo == 's',Thabitacion = 2)->elegircvhdoble(Presupuesto,Diash,Distancia,Tcomida,Clima);  
(Vehiculo == 'n',Thabitacion = 1)->elegirsvhsimple(Presupuesto,Diash,Distancia,Tcomida,Clima);  
(Vehiculo == 'n',Thabitacion = 2)->elegirsvhdoble(Presupuesto,Diash,Distancia,Tcomida,Clima)).
```

Dependiendo de las variables nos dirige a ciertas reglas donde se analizan los datos y se hacen matching con los datos que se encuentran en los hechos declarados al inicio del proyecto. Comparamos parámetros para ver si están dentro de los rangos permitidos por el usuario y para las expresiones aritméticas en un lado tenemos la variable seguido de is y de la operación aritmética.

```
svhdoble(Idioma,Dias,Distancia,Tcomida,Clima,Estrellas):-  
departamento(IDdep,Nombredep,_,Idiomah,Climadep),  
hotel(_,Nombreh,Direccion,Estrellash,_,Costohs,Costcomida,IDdep,Distanciah),  
  
CostoGasolina is 0.00,  
CostoComida is (Tcomida * Costcomida)*Dias,  
CostoHab is (Costohs * Dias),  
CostoHotelTotal is CostoGasolina+ CostoComida + CostoHab,  
  
Distanciah<Distancia,  
Climadep == Clima,  
Idiomah == Idioma,  
Estrellash >= Estrellas,  
  
resultadoidioma(Nombredep,Nombreh,Direccion,CostoGasolina,CostoHab,CostoHotelTotal,CostoComida),  
fail.
```

Cada una de estas reglas, en un lenguaje normal funcionaria como un método con parámetros que va llamando a otro método dentro de el para poder mostrar en pantalla la información obtenida del análisis de la información.

Se realizaron caminos especificando un filtro determinado como el clima, el idioma, el presupuesto y la cantidad de estrellas en el hotel.

El match se realiza solo utilizando los hechos de departamento y hotel, utilizando el id para poder relacionarlos.



```
resultadoidioma(Nombredep, Nombreh, Direccion, CostoGasolina, Costohs, CostoHotelTotal, CostoComida):-
nl,
write('-----'),
nl,
format('Esta opcion se ajusta a sus criterios
Hotel: ~a
Departamento: ~a
Direccion: ~a
Gasolina: ~a
Habitacion: ~a
Alimentacion: ~a
Total: ~a', [Nombreh, Nombredep, Direccion, CostoGasolina, Costohs, CostoHotelTotal, CostoComida]),
fail.
```

Reportes:

Como se puede ver necesitamos los hechos para poder obtener los hechos declarados anteriormente, es importante tener en cuenta el registro ya que permitirá relacionar el cliente con el hotel. Si en algún momento se necesita alguno de los otros hechos como el departamento es importante entablar la relación por medio del id para establecer la relación.

Para poder limitar el reporte según lo solicitado se realiza una comparación con operadores lógicos, en la imagen de arriba podemos extraer solo los que cumplan con que la opinión sea mayor o igual a 5 y por ultimo procedemos a realizar el print de los resultados.



Conclusiones

Esta herramienta brinda soporte para la generación de sistemas en la resolución de problemas con base a hechos y predicados.

Prolog es una herramienta que brinda grandes posibilidades de potencializar el pensamiento lógico.

Prolog sirve de base para generar sistemas expertos en inteligencia artificial.

En el desarrollo del proyecto se tiene una base de conocimiento, siendo capaz de analizar y comparar los datos para dar la mejor solución a las interrogantes de viaje.

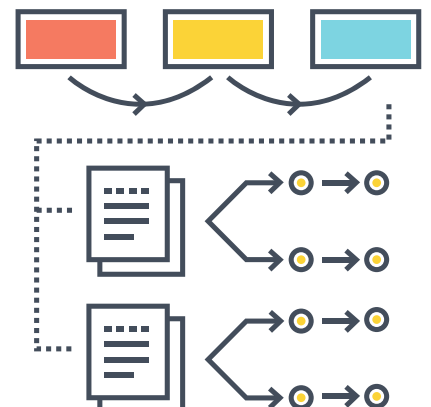
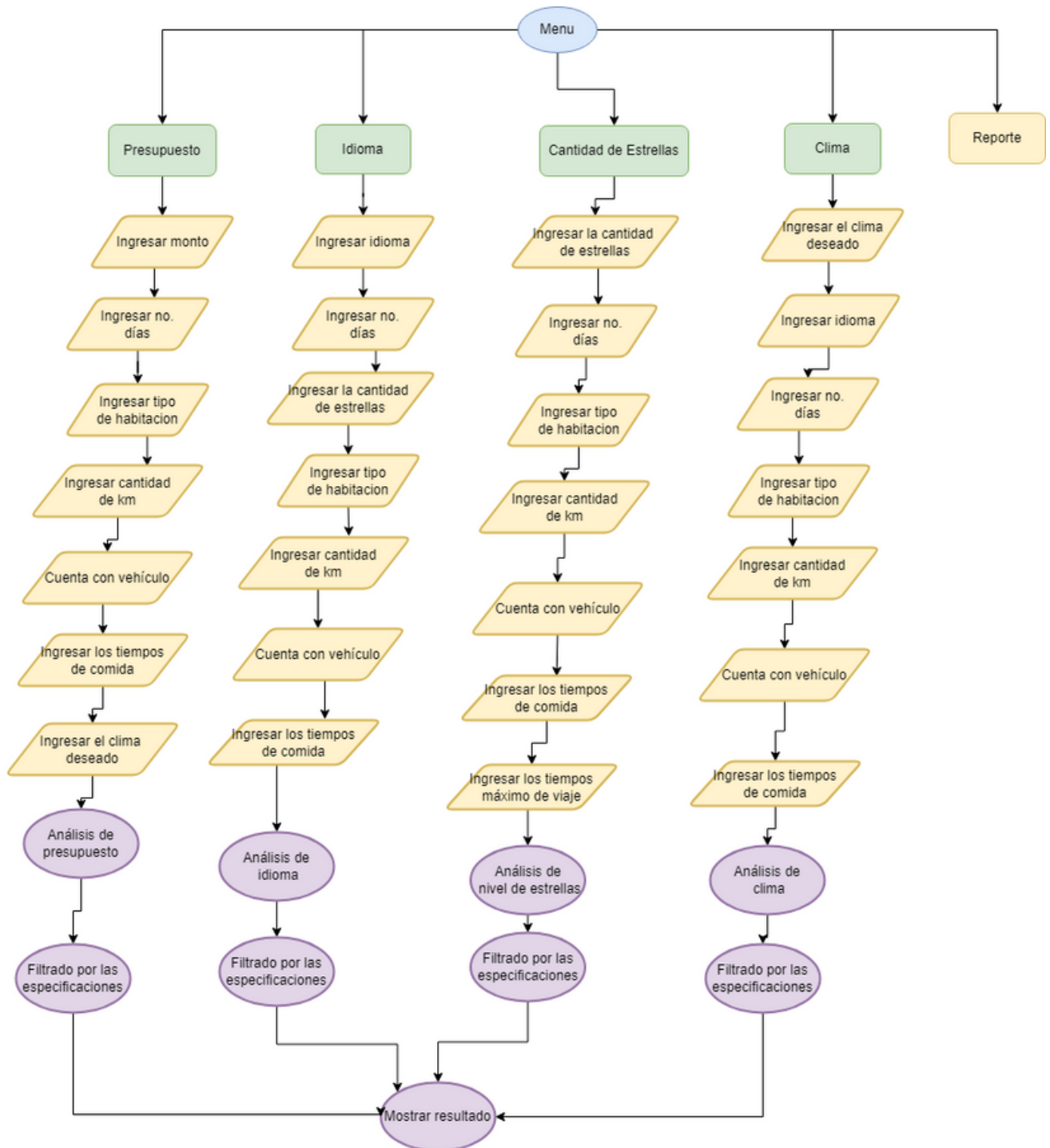


Diagrama de Encadenamiento





Explicación del diagrama

Se utilizaron diversas formas para diferenciar los procesos de los datos que el usuario ingresaba para poder realizar el flujo de cada uno de los caminos que conforman la solución.

A lo largo del proyecto se tienen una serie de 4 caminos que dependerán de la sección según los datos ingresados del usuario. Cada uno tiene una especificación que prevalece entre las demás, al finalizar de obtener todos los datos, se realiza el análisis ya sea de presupuesto, comida, etc.

Al finalizar el análisis, se realiza un ultimo filtrado con las opciones ingresadas como cantidad de km a recorrer, el clima u otro. Para que finalmente se muestren los resultados

