
Universidad de san Carlos de Guatemala

Facultad de ingeniería

Escuela de Ciencias y Sistemas

Inteligencia Artificial 1

Laboratorio

Ing. Luis Espino

Aux. Ana Isabel Culajay Gonzalez



Creación de Sistema Experto en Prolog

Katerine Adalinda Santos Ramírez

201908321

Guatemala, 20 de junio de 2023

OBJETIVO GENERAL

Proporcionar soluciones de un sistema experto basado en reglas el cual pueda tomar la decisión de indicar que hotel puede ser la mejor opción para hospedarse según las necesidades del usuario.

OBJETIVOS ESPECÍFICOS


1. Completar la comprensión de la programación lógica
2. Aplicar los conocimientos adquiridos del uso del lenguaje Prolog
3. Que el alumno pueda aplicar su ingenio en la realización de la solución del problema planteado
4. Que el alumno pueda observar las ventajas de la realización de un sistema experto como paso fundamental a una inteligencia artificial funcional
5. Comprender el uso y alimentación de la base de conocimientos para el uso del sistema experto

DETALLE DE LA CANTIDAD DE RECURSOS QUE UTILIZA EL SISTEMA EXPERTO EN EJECUCIÓN

Cuando se está compilando el programa se necesita

- **CPU:** 0%
- **Memoria Ram:** 19.9 MB
- **Disco:** 0 MB/s
- **Network:** 0 Mbps

Processes

^		4%	39%	2%	1%
Name	Status	CPU	Memory	Disk	Network
Apps (3)					
>  swipl-win (2)		0%	19.9 MB	0 MB/s	0 Mbps

DETALLE DE LA EXPLICACIÓN DEL CÓDIGO DE LA SOLUCIÓN

1. Se crearon todos los hechos los cuáles se extrajeron de una base de datos que estaba en access. Los valores se crean en base a hechos y valores. Ejemplo:

hecho(valor1,valor2)

```
1 % -----
2 % ----- BASE DE DATOS DE CONOCIMIENTO -----
3 % -----
4
5 % -----
6 % ----- CLIENTE -----
7 % -----
8 cliente(1,'Jose','Morales','Guatemala',27,'Soltero','Vacaciones').
9 cliente(2,'Fabrizio','Sartini','Italia',25,'Casado','Vacaciones').
10 cliente(3,'Gustavo','Molina','Holanda',15,'Soltero','Trabajo').
11 cliente(4,'Maria','Gutierrez','Guatemala',20,'Soltero','Vacaciones').
12 cliente(5,'Cornelio','Bustamante','Guatemala',18,'Soltero','Trabajo').
13 cliente(6,'Loraine','Cortez','Española',34,'Divorciado','Paso').
14 cliente(7,'Paco','Estanley','Guatemala',30,'Soltero','Vacaciones').
15 cliente(8,'Daniela','Samayoa','Guatemala',21,'Casado','Trabajo').
16 cliente(9,'Silvia','Santis','El Salvador',50,'Casado','Paso').
17 cliente(10,'Luis','Ortega','El Salvador',35,'Soltero','Vacaciones').
18 cliente(11,'Tatiana','Lopez','El Salvador',28,'Soltero','Paso').
19 cliente(12,'Ronny','Paiz','Honduras',20,'Soltero','Trabajo').
20 cliente(13,'Cesar','Ortiz','Guatemala',36,'Soltero','Paso').
21 cliente(14,'Margot','Claus','USA',32,'Soltero','Vacaciones').
22 cliente(15,'Jorge','Calderon','Venezuela',26,'Casado','Paso').
23 cliente(17,'Maira','Martinez','Mexico',25,'Soltero','Trabajo').
24 cliente(18,'Jazmin','Dolores','Venezuela',27,'Casado','Vacaciones').
25 cliente(19,'Mauro','Hernandez','Español',22,'Soltero','Vacaciones').
26 cliente(20,'Luis','Sagastume','Costa Rica',20,'Soltero','Trabajo').
27 cliente(21,'Rcharson','Patric','Inglaterra',30,'Casado','Vacaciones').
28 cliente(22,'Montheri','Lois','Inglaterra',32,'Casado','Vacaciones').
29 cliente(23,'Silverio','Perez','Mexico',33,'Casado','Paso').
30 cliente(24,'Monica','Suarez','Mexico',30,'Casado','Trabajo').
```

2. Se crea el menú principal, el cuál tiene dos opciones, “Planificar viaje” y “Reportes”. El usuario elegirá la que necesite en ese momento.

```
1 % -----
2 % ----- Menu inicio -----
3 % -----
4 inicio :- write('***** Bienvenido ***'), nl, nl,
5           write('Selecciona una opcion'), nl,
6           format('\t[1] Planificar viaje
7                 \t[2] Reportes
8                 -> '),
9           read(OP_INICIO), opcion_inicio(OP_INICIO).
10 opcion_inicio(OP_INICIO) :- (
11     OP_INICIO == 1 -> menu_criterios;
12     OP_INICIO == 2 -> menu_reportes
13 ).
```

3. Si se elige “Planificar viaje” se muestran las diferentes opciones para que el usuario seleccione a la que le da más importancia a la hora de planificar un viaje. Ya sea por el presupuesto, por el idioma, por la cantidad de estrellas del hotel o por el clima.

```
1 % -----
2 % ----- Planificacion -----
3 % -----
4
5 menu_criterios :- write('Selecciona el criterio mas importante'), nl,
6                   format('\t[1] Presupuesto
7                           \t[2] Idioma
8                           \t[3] Cantidad de estrellas del hotel
9                           \t[4] Clima
10                          -> '),
11                   read(OP_CRITERIO), opcion(OP_CRITERIO).
12
13 opcion(OP_CRITERIO) :- (
14     OP_CRITERIO == 1 -> presupuesto;
15     OP_CRITERIO == 2 -> idioma;
16     OP_CRITERIO == 3 -> estrellas;
17     OP_CRITERIO == 4 -> clima
18 ).
```

4. Si el usuario elige el flujo de presupuesto, este a su vez se separa en 3 flujos más, todos basándose en la cantidad de dinero que el usuario ingrese.

- **Alto presupuesto:** Monto de 3000 a cualquier monto superior
- **Mediano presupuesto:** Monto de 1000 a 2999
- **Bajo presupuesto:** Monto de 1 a 1000

```
1  % -----
2  % ----- Planificacion -----
3  % -----
4  presupuesto :- write('Cual es tu presupuesto?'),
5                read(PRESUPUESTO), tipos_presupuesto(PRESUPUESTO).
6
7  tipos_presupuesto(PRESUPUESTO) :- (
8      PRESUPUESTO < 1000 -> presupuesto_bajo(PRESUPUESTO);
9      PRESUPUESTO >= 3000 -> presupuesto_alto(PRESUPUESTO);
10     presupuesto_medio(PRESUPUESTO)
11 ).
```

5. Si se elige el flujo de bajo presupuesto, se muestra una serie de preguntas que servirán para encontrar el hotel que se adecue a las necesidades del usuario. Las opciones de alto presupuesto y mediano presupuesto son muy similares, la única diferencia son las diferentes preguntas que se le hacen al usuario.

Las diferentes preguntas que se le hacen al usuario se hacen a través de la función “write()” y las respuestas son capturadas por medio de la función “read()”.

```
1  presupuesto_bajo(PRESUPUESTO) :- write('** Plan de viaje economico **'), nl,
2      write('\tCual es tu presupuesto maximo a gastar en comida?: '),
3      read(COMIDA_MAX),
4      write('\tCual es tu presupuesto maximo a gastar en la habitacion?: '),
5      read(HABITACION_MAX),
6      write('\tCual es la puntuacion minima de los clientes? (0-10): '),
7      read(OPINION_MIN),
8      write('\tCuantas personas viajaran?: '),
9      read(CANTIDAD_PERSONAS),
10     write('\tQue tipo de habitacion prefieres? (simple, doble): '),
11     read(TIPO_HABITACION),
12     write('\tCuantos dias se hospedaran?: '),
13     read(CANTIDAD_DIAS),
14     write('\tCuantos tiempos de comida incluiras diarios? (0-3): '),
15     read(TIEMPOS_COMIDA),
16     write('\tViajas en vehiculo? (s/n): '),
17     read(VEHICULO),
18     inf_presupuesto_bajo(PRESUPUESTO, COMIDA_MAX, HABITACION_MAX, OPINION_MIN, CANTIDAD_PERSONAS, TIPO_HABITACION, CANTIDAD_DIAS, TIEMPOS_COMIDA, VEHICULO).
19
```

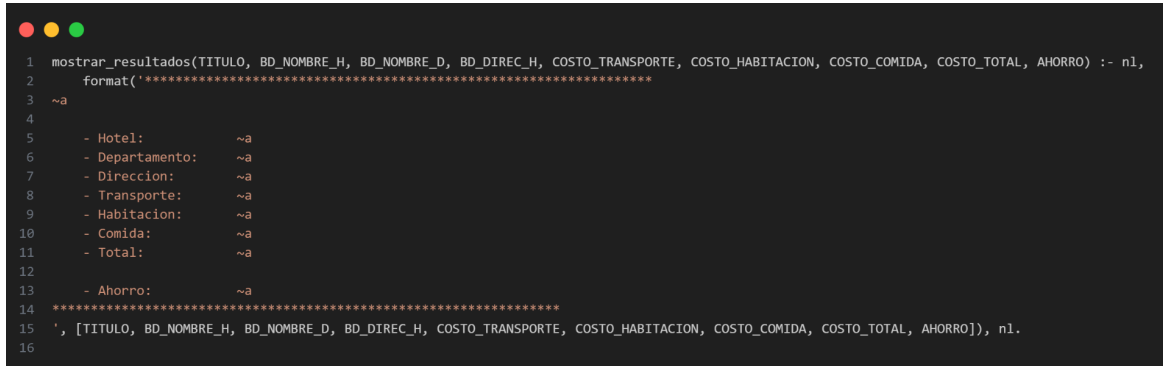
- Después de hacer las preguntas al usuario se llega a la sección de inferencia donde se decidirá qué hotel ofrecer según las respuestas obtenidas

Para realizar una inferencia se siguen los siguientes pasos.

- Se obtienen los valores necesarios de los hechos de tal manera que los datos que no sirven se colocan como “_”.
- Con los valores obtenidos se realizan las relaciones entre los hechos a través de sus valores similares tales como ID de hotel, ID del departamento, etc.
- Luego de relacionar los hechos se realizan los cálculos de transporte, habitación, comida, etc. Además de que se guardan en variables para luego utilizarlas.
- Luego se realizan las restricciones sobre los hechos que se obtuvieron hasta el momento, así se filtran solo los datos que son de interés. Cada restricción se realiza utilizando operadores de comparación como “=<”, “>=” y “==”.
- Al finalizar se realiza el cálculo del ahorro que el usuario tendrá en base al presupuesto dado y todos los datos se muestran como respuesta final en la sección de resultados.

```
1 % -----
2 % ----- REALIZAR INFERENCIAS -----
3 % -----
4 inf_presupuesto_bajo(PRESUPUESTO, COMIDA_MAX, HABITACION_MAX, OPINION_MIN, CANTIDAD_PERSONAS, TIPO_HABITACION, CANTIDAD_DIAS, TIEMPOS_COMIDA, VEHICULO) :- (
5     % BUSQUEDA DE HECHOS
6     hotel(ID_HOTEL, BD_NOMBRE_H, BD_DIREC_H, _, BD_HAB_SIMPLE_H, BD_HAB_DOBLE_H, BD_PRECIO_COMIDA_H, ID_DEPARTAMENTO, BD_DIS_H),
7     departamento(ID_DEPARTAMENTO, BD_NOMBRE_D, _, _, BD_PASAJE_D),
8     registro(_, _, ID_HOTEL, _, BD_OPINION_R),
9     % CALCULOS
10    (
11        VEHICULO == 's' -> COSTO_TRANSPORTE is (BD_DIS_H * 12.50 * 2);
12        VEHICULO == 'n' -> COSTO_TRANSPORTE is (BD_PASAJE_D * 2 * CANTIDAD_PERSONAS)
13    ),
14    COSTO_COMIDA is (BD_PRECIO_COMIDA_H * TIEMPOS_COMIDA * CANTIDAD_PERSONAS * CANTIDAD_DIAS),
15    (
16        TIPO_HABITACION == 'simple' -> COSTO_HABITACION is (BD_HAB_SIMPLE_H * CANTIDAD_PERSONAS * CANTIDAD_DIAS);
17        TIPO_HABITACION == 'doble' -> COSTO_HABITACION is (BD_HAB_DOBLE_H * CANTIDAD_PERSONAS * CANTIDAD_DIAS)
18    ),
19    COSTO_TOTAL is (COSTO_TRANSPORTE + COSTO_COMIDA + COSTO_HABITACION),
20    % RESTRICCIONES
21    COSTO_TOTAL <= PRESUPUESTO,
22    COSTO_COMIDA <= COMIDA_MAX,
23    COSTO_HABITACION <= HABITACION_MAX,
24    BD_OPINION_R >= OPINION_MIN,
25    % AHORRO Y RESULTADO
26    AHORRO is (PRESUPUESTO - COSTO_TOTAL),
27    mostrar_resultados('RESULTADO. CRITERIO PRESUPUESTO ECONOMICO', BD_NOMBRE_H, BD_NOMBRE_D, BD_DIREC_H, COSTO_TRANSPORTE, COSTO_HABITACION, COSTO_COMIDA, COSTO_TOTAL, AHORRO)
28 ).
```

-
7. La sección de mostrar resultados se encarga de recopilar todos los valores y así mostrarlos al usuario a través de la función `format()`, la cual permite realizar impresiones de datos de manera sencilla. Los valores “~a” dentro de la cadena, hacen referencia a mostrar un valor en ese lugar, los valores son colocados como parámetros en la función `format()` dentro de un arreglo tal como se muestra en la imagen.

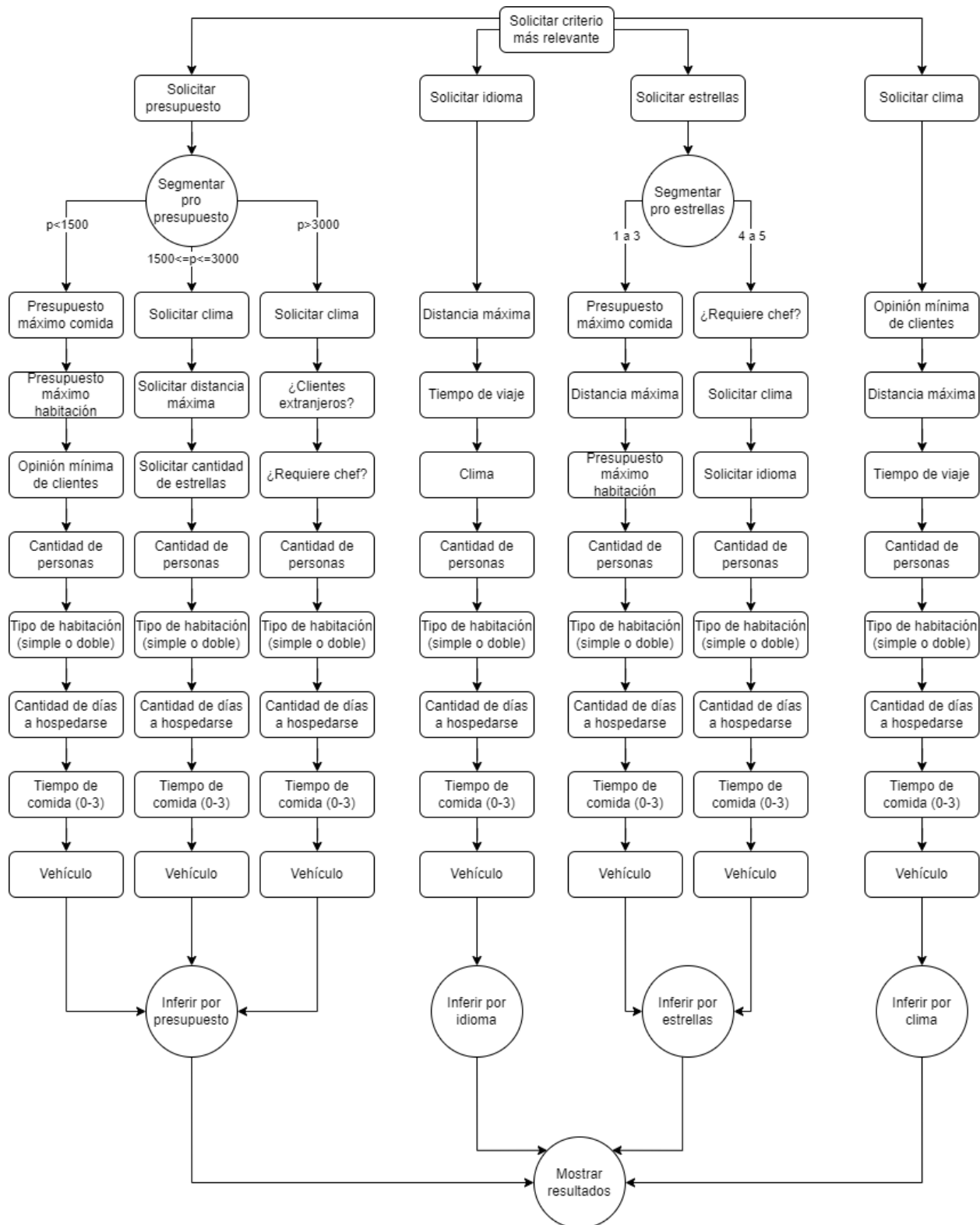


```
1  mostrar_resultados(TITULO, BD_NOMBRE_H, BD_NOMBRE_D, BD_DIREC_H, COSTO_TRANSPORTE, COSTO_HABITACION, COSTO_COMIDA, COSTO_TOTAL, AHORRO) :- nl,
2      format('*****
3      ~a
4
5      - Hotel:          ~a
6      - Departamento:  ~a
7      - Direccion:     ~a
8      - Transporte:    ~a
9      - Habitacion:    ~a
10     - Comida:        ~a
11     - Total:         ~a
12
13     - Ahorro:        ~a
14     *****
15     ', [TITULO, BD_NOMBRE_H, BD_NOMBRE_D, BD_DIREC_H, COSTO_TRANSPORTE, COSTO_HABITACION, COSTO_COMIDA, COSTO_TOTAL, AHORRO]), nl.
16
```

CONCLUSIONES

- El lenguaje de prolog lógico y declarativo ofrece flexibilidad para expresar reglas y relaciones complejas.
- En base a un sistema experto, podemos observar la manera en como un programa empieza a generar características de inteligencia artificial, demostrando su habilidad de inferencia en base a ciertos valores.
- La comprensión de un lenguaje de programación lógico tiene una alta curva de aprendizaje al tener que cambiar de un paradigma funcional o procedimental.
- Un sistema experto puede llegar a ser una herramienta potente, de fácil manipulación y filtrado para realizar todo tipo de consultas relacionales específicamente.
- La creación de reglas y hechos dentro de los lenguajes de programación lógica son el fundamento y la base del programa, forman parte de la base de conocimiento.

DIAGRAMA DE ENCADENAMIENTO



EXPLICACIÓN DEL DIAGRAMA DE ENCADENAMIENTO

El diagrama de encadenamiento mostrado arriba muestra de manera gráfica los diferentes flujos que puede llegar a tomar la decisión de cada usuario.

Los caminos principales del sistema experto se basan según cuatro criterios que son:

- Presupuesto
- Idioma
- Estrellas
- Clima

En cada uno de los diferentes flujos se muestra de una forma más detallada los datos que se le solicitarán al usuario para que así a partir de sus respuestas se pueda gestionar la información e inferirla posteriormente, así como mostrarle el resultado final al usuario donde cada flujo converge en el mismo estado final.