

9 DE OCTUBRE DE 2022

TAREA PROGRAMADA 2

GESTIÓN DE VEHÍCULOS EN SWI PROLOG

JEAN HUNT

LENGUAJES DE PROGRAMACIÓN
Ingeniería en computación




Tabla de contenido

<i>Manual de usuario</i>	1
Compilación y ejecución	2
Tutorial	3
1. Base de conocimiento	3
2. Relacionar	3
3. Es eco-amigable	4
4. Producción total	5
5. Es final, base o intermedio	5
6. Relacionados	7
7. Es conglomerado	7
8. Muestra vehículo	8
<i>Documentación general</i>	9
1. Descripción del problema	10
2. Diseño del programa:	10
2.1. <i>Decisiones de diseño</i>	10
2.2. <i>Algoritmos/predicados usados</i>	11
3. Librerías utilizadas	11
4. Análisis de resultados	12

Ilustración 1: Compilación y ejecución.....	3
Ilustración 2: Base de conocimiento – vehículos	3
Ilustración 3: Base de conocimiento – relaciones.....	4
Ilustración 4: esEcoAmigable	4
Ilustración 5: producción total	5
Ilustración 6:esBase.....	6
Ilustración 7:esIntermedio	6
Ilustración 8:esFinal	6
Ilustración 9: relacionados	7
Ilustración 10: esConglomerado	7
Ilustración 11:muestraVehiculo	8
Ilustración 12:Predicados utilizados.....	11
Ilustración 14:Objetivos alcanzados.....	12

Manual de usuario

Compilación y ejecución

Este programa de gestión de vehículos funciona a través de la consola de comandos, no utiliza ninguna librería externa al lenguaje por lo que lo único que se necesita para ejecutar el programa es la implementación SWI Prolog.

Esta implementación puede ser descargada e instalada desde el siguiente enlace: <https://www.swi-prolog.org/download/stable> , para el desarrollo de esta tarea se utilizó la versión 8.4.3-1 para Windows

Una vez instalado, para ejecutar el programa seguiremos los siguientes pasos:

1. Abrir una terminal de comandos, ya sea integrada o de terceros.
2. Dirigirse a la ruta en donde se encuentra el archivo proyecto.
3. Ejecutar swi prolog haciendo uso del siguiente comando:

swipl

4. Consultar el programa haciendo uso del siguiente comando:

consult("nombreArchivo.pro").

en este caso:

consult("main.pro").

Nota: Puede iniciar el programa desde cualquier directorio, pero tendrá que ingresar la ruta completa del archivo, por ejemplo:

consult("C://escritorio/programa/main.pro").

5. Todos los hechos del programa estarán listos para ser ejecutados.

```
C:\Users\jeanp\Desktop\TEC\2022\2[Semestre]\Lenguajes\Tareas\T2\JeanHunt\T2_lenguajes_SII_2022>swipl
Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

1 ?- consult("main.pro").
true.

2 ?-
```

Ilustración 1: Compilación y ejecución

Todos los datos que sean solicitados deberán ser ingresados mediante la consola de comandos, por ejemplo, argumentos de un hecho o entrada por teclado de algunos valores.

Tutorial

1. Base de conocimiento

El programa contiene una serie de vehículos registrados según su fabricante que son los que se utilizarán en los distintos cálculos y hechos. Estos vehículos se muestran a continuación:

```
nuevoVehiculo(toyotaGroup,vehiculo(toyota, hilux2021, pickup, 2500, 700)).
nuevoVehiculo(toyotaGroup,vehiculo(toyota, hilux2022, pickup, 3000, 750)).
nuevoVehiculo(fca,vehiculo(jeep, compass2019, cross, 2000, 450)).
nuevoVehiculo(fca,vehiculo(jeep, compass2020, cross, 2000, 450)).
nuevoVehiculo(fca,vehiculo(maserati, levante2021, cross, 2000, 450)).
nuevoVehiculo(fca,vehiculo(maserati, levante2022, cross, 1500, 650)).
nuevoVehiculo(nissanMC,vehiculo(nissan, sentra2012, sedan, 1600, 500)).
nuevoVehiculo(nissanMC,vehiculo(nissan, sentra2013, sedan, 1600, 500)).
nuevoVehiculo(nissanMC,vehiculo(nissan, almera2014, sedan, 1500, 550)).
nuevoVehiculo(nissanMC,vehiculo(nissan, almera2015, sedan, 1600, 550)).
nuevoVehiculo(nissanMC,vehiculo(nissan, sentra2016, sedan, 1800, 500)).
```

Ilustración 2: Base de conocimiento – vehículos

2. Relacionar

Los vehículos de la base de conocimiento pueden estar relacionados entre sí, es decir, un modelo específico puede tener un modelo siguiente, y por lo tanto un anterior. Estas

relaciones están dadas según fabricante, sin embargo, podría existir una relación entre vehículos de distintos fabricantes.

Una relación se define de la siguiente manera:

```
relacionar(toyotaGroup, hilux2021, hilux2022).
relacionar(fca, compass2019, compass2020).
relacionar(fca, compass2020, levante2021).
relacionar(fca, levante2021, levante2022).
relacionar(nissanMC, sentra2012, sentra2013).
relacionar(nissanMC, sentra2013, almera2014).
relacionar(nissanMC, almera2014, almera2015).
relacionar(nissanMC, almera2015, sentra2016).
```

Ilustración 3: Base de conocimiento – relaciones

3. Es eco-amigable

Un auto es eco amigable si cumple con alguna de las siguientes condiciones:

- Es sedan con cilindraje menor a 2000 y autonomía mayor a 500.
- Es cross con cilindraje menor a 2500 y autonomía mayor a 450.
- Es pickup con cilindraje menor a 3000 y autonomía mayor a 650.

Para ejecutar este predicado se debe indicar el nombre del fabricante y modelo, por ejemplo:

```
9 ?- esEcoAmigable(toyotaGroup, hilux2021).
Estilo: pickup   Cilindraje: 2500   Autonomia: 700
Es eco amigable
true .

10 ?- esEcoAmigable(toyotaGroup, hilux2022).
Este vehiculo no existe o no es eco amigable
false.
```

Ilustración 4: esEcoAmigable

En la imagen anterior se muestran los dos posibles casos, la consulta 9 corresponde a una sentencia que se cumple, es decir, el modelo hilux2021 es eco-amigable, caso contrario de la consulta 10 en la que el hilux2022 no es eco-amigable.

4. Producción total

Este predicado permite calcular cuantos autos hay asociados a un fabricante en específico o si la cantidad indicada es la correcta. Para ello se debe utilizar el predicado de la siguiente manera *produccionTotal(nombreFabricante, X)*, en lugar de X se puede usar cualquier otro átomo siempre y cuando sea diferente del nombre del fabricante y no sea un número.

```
17 ?- produccionTotal(nissanMC, X).
X = 5.

18 ?- produccionTotal(nissanMC, Cantidad).
Cantidad = 5.

19 ?- produccionTotal(nissanMC, 5).
true.

20 ?- produccionTotal(nissanMC, 4).
false.
```

Ilustración 5: producción total

En la imagen anterior los ejemplos 17 y 18 corresponden al calculo de la cantidad total de vehículos, vemos como el segundo átomo del predicado será el que indique la respuesta. Los ejemplos 19 y 20 corresponden a la verificación de la cantidad de autos, el predicado será falso si el segundo átomo es diferente de la cantidad de vehículos.

5. Es final, base o intermedio

Este conjunto de funciones permite saber si un vehículo es base, final o intermedio, estos términos corresponden a:

- Final es cuando no tiene vehículos siguientes.
- Base cuando no tiene anteriores.
- Intermedio tiene anterior y final.

Los predicados correspondientes a estas funciones reciben el nombre del fabricante y del modulo tal y como se muestra en las siguientes llamadas:

```
21 ?- esBase(nissanMC, sentra2012).
Es base
true.

22 ?- esBase(nissanMC, sentra2013).
false.
```

Ilustración 6:esBase

```
24 ?- esIntermedio(nissanMC, almera2014).
Es intermedio
true.

25 ?- esIntermedio(nissanMC, sentra2012).
false.
```

Ilustración 7:esIntermedio

```
31 ?- esFinal(nissanMC, sentra2016).
Es final
true.

31 ?- esFinal(nissanMC, sentra2015).
false.
```

Ilustración 8:esFinal

Las llamadas 21 y 22 corresponden al predicado *esBase*, 24 y 25 a *esIntermedio*, 31 y 32 a *esFinal*. En todos los casos un ejemplo falso y otro verdadero.

6. Relacionados

Este predicado permite saber si dos vehículos están relacionados, es decir, tienen un camino entre sí. Estas relaciones se dan entre vehículos de un mismo fabricante. Las llamadas al predicado lucen de la siguiente manera:

```
36 ?- relacionados(nissanMC, sentra2012, almera2014).
Estan relacionados y esta es su ruta: sentra2012 -> sentra2013 -> almera2014
true .

37 ?- relacionados(nissanMC, sentra2016, sentra2013).
Estan relacionados y esta es su ruta: sentra2013 -> almera2014 -> almera2015 -> sentra2016
true .

38 ?- relacionados(nissanMC, sentra2016, hilux2022).
No estan relacionados.
false.
```

Ilustración 9: relacionados

En donde 36 y 37 representan relaciones en ambas direcciones y 38 la ausencia de relación.

7. Es conglomerado

Este predicado permite identificar si un fabricante es conglomerado o no. Un fabricante es conglomerado si fabrica más de una marca. La ejecución de este predicado se muestra en la siguiente imagen:

```
39 ?- esConglomerado(fca).
Es conglomerado
true .

40 ?- esConglomerado(toyotaGroup).
No es conglomerado
true .
```

Ilustración 10: esConglomerado

El ejemplo 39 corresponde a un fabricante conglomerado y el ejemplo 40 a un fabricante no conglomerado.

8. Muestra vehículo

Este predicado muestra toda la información de un modelo indicado, por ejemplo:

```
56 ?- muestraVehiculo(nissanMC, sentra2016).
----- Informacion del vehiculo -----

Fabricante: nissanMC
Marca: nissan
Modelo: sentra2016
Estilo: sedan
Cilindraje: 1800
Autonomia: 500
-----
true.

57 ?- muestraVehiculo(nissanMC, sentra2018).
El vehiculo no existe
false.
```

Ilustración 11:muestraVehiculo

En el ejemplo 56 el vehículo existe por lo tanto se imprime la información de dicho modelo, sin embargo, en el ejemplo 57 el vehículo no existe.

Documentación general

1. Descripción del problema

Se busca desarrollar una serie de casos, para los cuáles se deberán generar hechos, predicados, reglas y unificación utilizando SWI Prolog.

Un conglomerado automotriz quiere organizar su producción de vehículos de las diferentes marcas que representan y estilos para públicos meta. Por lo que se deberán generar predicados que permitan insertar, relacionar y consultar información sobre dichos vehículos y fabricantes.

***Nota:** La descripción anterior fue extraída de la especificación correspondiente a esta tarea programada

2. Diseño del programa:

El programa fue desarrollado en el lenguaje de programación SWI Prolog. La interfaz del programa será únicamente a través de la consola de comandos y mediante entrada del teclado.

El programa se compone de un único archivo fuente:

- **Main.pro:** Contiene todos los predicados con las funcionalidades relacionadas al sistema de gestión de vehículos solicitado.

2.1. Decisiones de diseño

Antes y durante el desarrollo del proyecto se tomaron las siguientes decisiones:

- El programa será desarrollado en lenguaje SWI Prolog.
- Se utilizará vscode como entorno de desarrollo.

- Toda llamada a los predicados se deberá hacer mediante consola
- El programa se desarrollará únicamente para ser ejecutado en consola.
- Únicamente se utilizarán librerías ya incluidas en el lenguaje en caso de ser necesarias.
- Se utilizará github como control de repositorio.

2.2. Algoritmos/predicados usados

En la siguiente imagen, se encuentra un listado de los predicados definidos en el programa:

```

1  esEcoAmigable(Fabricante, Modelo):-
2  esEcoAmigableAux(Fabricante, Modelo):-
3  produccionTotal(Fabricante, X):-
4  esFinal(Fabricante, Modelo):-
5  esBase(Fabricante, Modelo):-
6  esIntermedio(Fabricante, Modelo):-
7  relacionados(Fabricante, Modelo1, Modelo2):-
8  relacionadosAux(Fabricante, Modelo1, Modelo2, Res):-
9  calcularRelacion(Fabricante, Modelo1, Modelo2,S,Res) :-
10 esConglomerado(Fabricante):-
11 esConglomeradoAux(Fabricante, Marca1):-
12 muestraVehiculo(Fabricante, Modelo):-
13 validarFabricante(Fabricante):-

```

Ilustración 12: Predicados utilizados

3. Librerías utilizadas

Para el desarrollo del programa no fue necesario el uso de ninguna librería extra o externa al lenguaje ya que solo se utilizaron funcionalidades base de este.

4. Análisis de resultados.

En la siguiente tabla se muestra una lista de objetivos para los cuales se evaluará si fueron cumplidos o no, en caso de no cumplirse también se indicará la razón del porqué.

Objetivo	Alcanzado	Razón
Adquirir conocimientos sobre el paradigma lógico	Si	
Cumplir con las funcionalidades del programa solicitadas	Si	
Obtener una alta rigidez durante la ejecución	Si	
Documentación interna del programa	Si	
Documentación externa del programa	Si	

Ilustración 13:Objetivos alcanzados

En el siguiente enlace se puede encontrar el repositorio que contiene la tarea
https://github.com/JPHuntV/T2_lenguajes_SII_2022.git