



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Facultat d'Informàtica de Barcelona



DIAGNOSTICADOR D'AVERIES EN COTXES I CUINES

GRAU EN INTEL·LIGÈNCIA ARTIFICIAL



SWI Prolog

Alumnes: Óscar Lu i Chenhui Lucía Wu

Professor: Ramón Sangüesa Sole, Caroline König

Assignatura: Coneixement i Raonament Automàtic

ÍNDEX

INTRODUCCIÓ	2
EXPLICACIÓ DEL PROGRAMA	3
SISTEMA COTXE	4
SISTEMA CUINA	7
SISTEMA DE DIAGNÒSIS.....	8
MANUAL D'USUARI PER AL SISTEMA DE DIAGNÒSTIC D'AVERIES	26
RESULTATS.....	29
CONCLUSIONS	36
OBSERVACIONS.....	37
BIBLIOGRAFIA	38

INTRODUCCIÓ

En aquest projecte hem desenvolupat un sistema en *Prolog* dissenyat per diagnosticar i explicar avaries en diversos subsistemes, aplicant **un raonament basat en regles i en les relacions causals entre símptomes i possibles causes**. Aquesta pràctica se centra en la detecció d'avaries en els àmbits dels automòbils i les cuines, complint els requisits establerts per als diferents nivells d'ambició del projecte, **nosaltres em escollit el nivell màxim d'ambició**.

El sistema permet la identificació interactiva d'avaries mitjançant preguntes dirigides a l'usuari, i, basant-se en les respostes, ofereix una llista de possibles causes. A més, proporciona explicacions raonades que vinculen cada causa amb els símptomes detectats. Una de les característiques més destacades del sistema és la seva flexibilitat, que facilita el treball amb diversos àmbits i permet carregar fitxers de coneixement o casos observats per ampliar-ne l'abast.

Els objectius principals d'aquest treball han estat:

1. **Desenvolupar un sistema eficient per al diagnòstic i l'explicació d'avaries**, basat en un model de relacions causa-efecte (no hem servit abducció).
2. **Implementar un programa interactiu** que permeti a l'usuari introduir observacions i obtenir explicacions precises.
3. **Dissenyar una arquitectura flexible i ampliable** que separi el motor de diagnosi dels dominis específics, com els automòbils o les cuines.
4. **Assegurar que el projecte compleix els requisits establerts per a l'avaluació**, seguint criteris de qualitat i bones pràctiques tant en el desenvolupament del sistema com en la seva documentació.

Aquest informe s'explicarà el codi, el seu funcionament i es presentaran exemples de sortida. Finalment, es detallaran algunes de les dificultats trobades durant el desenvolupament del projecte i es conclourà amb una reflexió final.

EXPLICACIÓ DEL PROGRAMA

En aquest apartat explicarem els diferents fitxers creats per desenvolupar aquest programa:

- Fitxer amb predicats sobre el sistema de cotxes: **cotxe.pl**
- Fitxer amb predicats sobre el sistema de cuina: **cuina.pl**
- Fitxer del sistema de diagnosi: **SolucioOscarLucia.pl**

Per la formalització dels predicats dels fitxers cotxe.pl i cuina.pl, hem buscat informació a les webs (citades a la bibliografia), i com podreu veure, surten problemes mes comuns de cada subsistema, causes i conseqüències. El que hem fet ha sigut simplement relacionar-les entre elles dins dels fitxers.

En quan al fitxer del diagnòstic, a part d'anar provant, també hem cercat per internet comandes que no coneixíem, també dir, tal i com posa a la introducció, hem utilitzat un raonament basat en relacions causals entre símptomes i possibles causes en comptes de l'abducció. Més endavant s'explica el funcionament del fitxer diagnòstic.pl.

Recomanem tenir els fitxers i el programa a l'abast, ja que així s'entendrà tot molt millor.

També cal esmentar que el programa està desenvolupat en castellà, sense cap raó específica, i esperem que això no generi cap confusió.

SISTEMA COTXE

El fitxer **cotxe.pl** conté una base de dades en Prolog que modela el domini dels cotxes, centrant-se específicament en els **cinc subsistemes, direcció, transmissió, encesa, sistema elèctric i confort interior**. Aquests són els subsistemes més generals que poden presentar avaries (o almenys els que es van proposar a les instruccions de la pràctica), les possibles causes d'aquestes avaries i les relacions entre causes i efectes observats. Aquest fitxer permet identificar problemes en els cotxes a partir de les evidències observades.

Directiva disjointa

```
:- disjointa averia/2.
:- disjointa causa/2.
:- disjointa relacion/2.
```

S'utilitza la directiva **:- disjointa/1** per especificar que certs predicats, en aquest cas, **avaria/2, causa/2 i relacion/2** poden tenir les seves clàusules distribuïdes de manera no contiguous al llarg del codi.

Predicats principals

subsistema/2: Aquest predicat indica els subsistemes que estan en bon funcionament dins del cotxe. La seva forma és la següent:

```
subsistema(coche, Subsistema).
```

avaria/2: Aquest predicat defineix les possibles avaries o efectes observables en cada subsistema del cotxe. La seva forma és la següent:

```
avaria(Subsistema, Avaria).
```

causa/2: Aquest predicat especifica les possibles causes de fallades per a cada subsistema. La seva forma és la següent:

```
causa(Subsistema, Causa).
```

relacion/2: Aquest predicat establirà entre una causa específica i els efectes esperats que aquesta pot produir. La seva forma és la següent:

```
relacion(Causa, EfectosEsperados).
```

Subsistemes del Cotxe

El fitxer defineix cinc subsistemes principals del cotxe, cadascun amb les seves possibles avaries i causes associades:

Direcció:

- **Avaries:** Vibracions del volant, soroll al girar, direcció dura, el cotxe no gira, desviació cap a un costat.
- **Causes:** Falta de líquid de direcció, fallades en la bomba de direcció assistida, fallada en les ròtules, fallada en els bujes, desgast als braços de control, pressió inadequada, corretja de la direcció fluixa, falta de lubricació en el sistema de ròtules, falta de lubricació en el sistema de cremallera, rodes mal alineades, desgast de pneumàtics, problemes de pressió dels pneumàtics, suspensió danyada, deterioració dels silentblocks, problemes en la barra de direcció, trencament de peces.

Transmissió:

- **Avaries:** Dificultat per canviar marxes, pèrdua de potència, sorolls estranys en la caixa, vibracions en conduir, sobreescalfament.
- **Causes:** Nivell baix d'oli de transmissió, desgast de sincronitzadors, embragatge desgastat, filtre obstruït, fallada en la bomba d'oli, convertidor de parell defectuós, engranatges desgastats, suports del motor desgastats.

Encesa:

- **Avaries:** El motor no arrenca, ralenti irregular, consum excessiu de combustible, detonacions, pèrdua de potència, espurna feble.
- **Causes:** Bugies desgastades, cables de bugies danyats, bobina d'encès fallida, fallada del sensor de cigonyal, distribuïdor defectuós, mescla de combustible pobre, regulador de voltatge defectuós.

Sistema Elèctric:

- **Avaries:** Llums que no funcionen, bateria descarregada, alternador fallit, curtcircuits, fusibles fosos, sistema que no respon, sobreescalfament de cables.
- **Causes:** Cablejat danyat, bateria antiquada, regulador de voltatge defectuós, connexions soltes, alternador danyat, curtcircuit.

Sistema de Confort Interior:

- **Avaries:** Fallades en el sistema de climatització, problemes en el sistema d'àudio, dificultats per pujar i baixar les finestres, els seients no s'ajusten, el tancament centralitzat no funciona, les llums interiors no funcionen.
- **Causes:** Fugides de refrigerant, compressor defectuós, sensor de temperatura defectuós, obstrucció en les sortides d'aire, fusible fos, problema de cables, amplificador trencat, trencament del mecanisme de l'alçavidres, acumulació de brutícia, motor d'elevació no funciona, palanca del seient danyada, peces danyades, motor elèctric avariats, problema amb el solenoide.

SISTEMA CUINA

El fitxer **cuina.pl** té la mateixa estructura que **cotxes.pl**, té tres subsistemes: forns, refrigeradors i fogons. Com és un sistema addicional, hem decidit no crear un domini tan extens com el del fitxer **cotxes.pl**.

Els seus predicats són els mateixos que els del fitxer **cotxe.pl**, excepte en el predicat **subsistema/2**, on en comptes de cotxe, es fa servir cuina.

Subsistemes de cuina

El fitxer defineix tres subsistemes de cuina, cadascun amb les seves possibles avaries i causes associades:

Forn:

- **Avaries:** No escalfa, escalfa massa, fa soroll.
- **Causes:** Resistència cremada, termòstat fallit, ventilador trencat

Refrigerador:

- **Averies:** No refreda, fa soroll, fuga d'aigua.
- **Causes:** Compressor trencat, ventilador trencat, mànega trencada.

Fogons:

- **Averies:** No encenen, flama baixa, fuga de gas.
- **Causes:** Espurna trencada, regulador de gas fallit, mànega trencada.

SISTEMA DE DIAGNÒSIS

Aquest fitxer **SolucioOscarLucia.pl** és el programa encarregat de realitzar els diagnòstics. Està dissenyat per identificar causes probables a partir d'evidències, un cop s'ha importat un domini, com en el nostre cas: **cotxe.pl**

Directiva dynamic

```
:- dynamic preguntada/1.
:- dynamic respuesta/1.
:- dynamic causas_probables/1.
:- dynamic evidencia/1.
:- dynamic relacion/2.
:- dynamic sistema_cargado/1.
:- dynamic hecho_observado/1.
```

S'utilitza la directiva **:- dynamic/1** per declarar **predicats dinàmics**, és a dir, predicats els fets dels quals poden ser **afegits** o **eliminats** durant l'execució del programa, a diferència dels predicats estàtics que no es poden modificar un cop definits.

Predicat principal (main)

La regla **main/0** actua com a punt d'entrada principal per a la inicialització del sistema. Quan és invocada, esborrarà tota la informació acumulada de les execucions anteriors utilitzant la directiva **retractall/1**. Aquesta directiva esborra els fets relacionats amb preguntes (**preguntada/1**), respostes (**respuesta/1**), causes probables (**causas_probables/1**), evidències observades (**evidencia/1**) i els fets observats (**hecho_observado/1**).

Així, es garanteix que el sistema comenci en un estat net i lliure de qualsevol dada prèvia, evitant possibles conflictes i assegurant que la nova execució sigui independent de les anteriors.

```
main :-
    retractall(preguntada(_)),
    retractall(respuesta(_)),
    retractall(causas_probables(_)),
    retractall(evidencia(_)),
    retractall(hecho_observado(_)),
```

La comanda **format/2** s'utilitza per generar sortides formatades en pantalla, permetent controlar l'aspecte visual del text que es vol mostrar. En aquest cas, la comanda següent s'utilitza per mostrar un missatge de benvinguda amb un estil decoratiu:

```
format('~n~`=t~60|~nBienvenido al sistema de diagnostico de
averias.~n~`=t~60|~n'),
```

El missatge es veurà així:

```
=====
Bienvenido al sistema de diagnostico de averias.
=====
```

Després, el sistema sollicita a l'usuari que introdueixi el nom del domini a carregar sense l'extensió **.pl**.

```
format('Cargue un sistema (ponga el nombre del archivo sin
extensión .pl): '),
```

El missatge es veurà així:

```
Cargue un sistema (ponga el nombre del archivo sin extensión .pl):
```

Amb la comanda **read/1** es llegeix el nom del fitxer que l'usuari introdueix per la consola i el nom es guarda en una variable anomenada **ArchivoSistema**.

```
read(ArchivoSistema),
```

Un cop tenim el nom del fitxer introduït, utilitzem la comanda **atom_concat/3** per concatenar l'extensió **.pl** amb el nom del fitxer i el guardem a la variable **RutaCompletaSistema**.

```
atom_concat(ArchivoSistema, '.pl', RutaCompletaSistema),
```

Després concatenem la ruta del directori amb el nom del fitxer. El primer argument serà la ruta del directori, el segon serà el nom del fitxer i el tercer serà el resultat de la concatenació:

```
atom_concat('ruta/del/directori/', ArchivoSistema, RutaSistema),
```

Aquesta línia concatenarà la ruta de la carpeta on es troba el fitxer amb el nom del fitxer, amb l'extensió **.pl**. Per exemple, si el directori és **'C:/Users/usuari/OneDrive/Escritorio/Practica/'** i el nom del fitxer és **'coche.pl'** o **'cocina.pl'**, el resultat de **RutaSistema** serà **'C:/Users/usuari/OneDrive/Escritorio/Practica/coche.pl'** (o **/cocina.pl**).

Amb el camí complet del fitxer, podem carregar-lo al sistema amb la comanda **consult/1**.

```
consult(RutaSistema),
```

Imprimim un missatge per pantalla indicant que la base de coneixement s'ha importat correctament després de carregar el domini.

```
format('Archivo ~w.pl cargado correctamente.\n', [ArchivoSistema]),
```

Es missatge es veurà així:

```
Archivo coche.pl cargado correctamente.
```

Després d'haver carregat el domini, el sistema li preguntarà a l'usuari si vol carregar els "fets observats" des d'un arxiu.

```
format('?Desea cargar hechos observados desde un archivo? (si/no) '),
```

El missatge es veurà així:

```
¿Desea cargar hechos observados desde un archivo? (si/no) |:
```

Aquesta línia espera que l'usuari escrigui una resposta a la consola, en aquest cas, **si**. o **no**.. El valor que l'usuari escriu es guarda en la variable **RespuestaHechos** fent servir la funció **read/1**.

```
read(RespuestaHechos),
```

Si el valor de **RespuestaHechos** és igual a **"si"**, s'executarà el predicat **cargar_hechos**. Aquest predicat s'encarregarà de carregar els fets observats des d'un arxiu.

```
( RespuestaHechos == si -> cargar_hechos
```

En canvi, si la resposta és “no”, s’executarà el predicat **iniciar**. Aquest predicat s’encarregarà d’inicialitzar el sistema per començar a fer preguntes a l’usuari per identificar les possibles causes de l’averia del cotxe.

```
; iniciar
),
```

Finalment, es troba el predicat **esperar_comando**, que s’executarà una vegada qualsevol dels procediments anteriors (**cargar_hechos** o **iniciar**) hagin finalitzat.

```
esperar_comando.
```

Carregar fets observats (cargar_hechos)

Aquest predicat s’encarrega de carregar els fets observats des d’un arxiu seleccionat per l’usuari.

```
cargar_hechos :-
```

El procediment és pràcticament idèntic al de carregar un sistema, ja que utilitza les mateixes comandes, com ara **read** per llegir el nom del fitxer i **consult** per carregar-lo.

```
format('Ingrese el nombre del archivo de hechos observados (sin
extension .pl): '),
read(Archivo),
atom_concat(Archivo, '.pl', RutaCompleta),
atom_concat('C:/Users/luosc/OneDrive/Escritorio/Practica/Paz/',
RutaCompleta, Ruta),
consult(Ruta),
```

Un cop finalitzada la càrrega, es mostra un missatge indicant que els fets observats s’han carregat correctament.

```
format('Hechos observados cargados desde ~w.pl~n', [Archivo]),
```

A continuació, es crida el predicat **registrar_hechos_observados**, que s’encarrega de registrar els fets observats com a respostes.

```
registrar_hechos_observados,
```

Finalment, es crida el predicat **iniciar_diagnostico**, que inicia el procés de diagnòstic directament, sense la necessitat de fer preguntes a l'usuari, ja que els fets observats han estat previamente carregats.

```
iniciar_diagnostico.
```

Registrar fets observats (registrar_hechos_observados)

Aquest predicat té com a funció registrar els fets observats dins del sistema. En el nostre cas, el fitxer que conté els fets són **hechos_coche.pl** (fets del cotxe) o **hechos_cocina.pl** (fets de la cuina). El seu procediment consta de tres passos principals.

```
registrar_hechos_observados :-
```

Primer, mitjançant la comanda **retractall(hecho_observado(_))**, elimina tots els fets observats que estiguessin prèviament registrats. Això assegura que només es tinguin en compte els nous fets que s'estan registrant en aquest moment.

```
retractall(hecho_observado(_)),
```

Utilitzant **findall(Efecto, respuesta(Efecto), EfectosObservados)**, es genera una llista de tots els efectes observats que han estat confirmats com a respostes afirmatives. Aquesta llista es guarda a la variable **EfectosObservados**.

```
findall(Efecto, respuesta(Efecto), EfectosObservados),
```

Efecto és una **variable** que representa qualsevol valor que podem obtenir de la base de fets quan executem el predicat **findall/3**. Per exemple:

Si al fitxer **hechos_coche.pl** tenim:

```
respuesta(vibraciones_volante).
```

```
respuesta(no(coche_no_gira)).
```

```
respuesta(ralentí_irregular).
```

Efecto podria ser un d'aquests fets: **vibraciones_volante**, **no(coche_no_gira)**, **ralentí_irregular**. Llavors, el predicat **respuesta/1** es fa servir per verificar si un **fet específic** és veritable a la base de fets.

Si féssim la consulta:

?- **respuesta(vibraciones_volante)** ens retornarà **true**.. Aquesta resposta **true** farà que el fet **vibraciones_volante** es guardi a la llista **EfectosObservados**. En canvi, **respuesta(no(coche_no_gira))** ens retornarà **false**. i, per tant, no es guardaria a la llista.

A continuació, utilitzem **forall/2** per iterar sobre cada element de la llista **EfectosObservados**. En cada iteració, es comprova si l'element (un efecte en concret) es troba a la llista, i si és així, s'afegeix a la base de dades de **hecho_observado/1** amb el predicat **assertz(hecho_observado(Efecto))**.

```
forall(member(Efecto, EfectosObservados),
assertz(hecho_observado(Efecto))).
```

El predicat **membre(Efecto, EfectosObservados)** comprova que cada **Efecto** de la llista **EfectosObservados** existeix en la llista.

Iniciar diagnòstic sense preguntes (iniciar_diagnostico)

Aquest predicat inicialitza el diagnòstic sense fer preguntes a l'usuari, utilitzant els fets observats carregats prèviament.

```
iniciar_diagnostico :-
```

Amb el predicat **retractall/1** s'esborra tots els fets relacionats amb les preguntes, les causes probables i les evidències anteriors. Això és necessari per evitar que la informació d'un diagnòstic anterior interfereixi amb el diagnòstic actual.

```
retractall(preguntada(_)),
retractall(causas_probables(_)),
retractall(evidencia(_)),
```

Seguidament, el predicat **findall/3** crea una llista **Efectos** amb tots els efectes observats que han estat carregats al sistema. Suposant que els efectes es van guardar en el fet **hecho_observado/1**.

```
findall(Efecto, hecho_observado(Efecto), Efectos),
```

Els predicats **forall/2** i **member/2** recorre tots els efectes observats de la llista **Efectos** i, per cada un, afegeix un fet a **evidencia(Efecto)** i el marca com a preguntat amb **preguntada(Efecto)**.

```
forall(member(Efecto, Efectos),
      (  assertz(evidencia(Efecto)),
        assertz(preguntada(Efecto))
      )
),
```

Un cop els efectes han estat afegits com a evidències i preguntats, es mostra un missatge que indica que el diagnòstic ha acabat, i també s'ofereixen opcions per veure les possibles causes i explicacions del diagnòstic.

```
format('Diagnostico completado.\n\nPuede solicitar las posibles causas
con "causas." y la explicacion con "explicacion."\n'),
```

Finalment, el predicat **esperar_comando/0** és cridat per permetre que l'usuari introdueixi un nou comandament, com "**causas.**" o "**explicacion.**" per obtenir més informació.

```
esperar_comando.
```

Iniciar Diagnòstic (iniciar)

Aquest predicat serveix per iniciar un diagnòstic demanant preguntes a l'usuari sobre els efectes observats, només quan no es carrega un fitxer de fets observats prèviament.

```
iniciar :-
```

Com els predicats anteriors, comença utilitzant el predicat **retractall/1** per eliminar qualsevol informació prèvia que pugui haver estat carregada. Això evita que el sistema barregi informació de diagnòstics anteriors amb el diagnòstic actual.

```
retractall(preguntada(_)),
retractall(causas_probables(_)),
retractall(evidencia(_)),
```

Amb la comanda **findall(Subsistema, subsistema(_, Subsistema), Subsistemes)** recull tots els subsistemes definits a la base de coneixement. La llista **Subsistemes** contindrà tots els subsistemes disponibles per a realitzar el diagnòstic.

```
findall(Subsistema, subsistema(_, Subsistema), Subsistemes),
```

Per exemple:

Suposem que tenim:

```
subsistema(coche, direccion).
subsistema(coche, transmission).
```

En aquest cas, la llista **Subsistemas** seria **[direccion, transmission]**.

Amb **forall(member(Subsistema, Subsistemas), ...)**, es recorre la llista de **Subsistemas** per dur a terme les següents accions per a cada subsistema:

- **Mostrar un títol per al subsistema:** Utilitzant **format**, es mostra un títol que indica el subsistema actual que es va a diagnosticar.

```
format('~n~`=t~60|~nPreguntas para el subsistema:
~w~n~`=t~60|~n', [Subsistema]),
```

- **Recollir els efectes de l'avaría.** A continuació, s'utilitza **findall(Efecto, averia(Subsistema, Efecto), Efectos)** per recollir tots els efectes associats al subsistema actual en una llista anomenada **Efectos**.

```
findall(Efecto, averia(Subsistema, Efecto), Efectos),
```

- **Fer preguntes sobre els efectes (si n'hi ha):** Si la llista **Efectos** no està buida, es crida al predicat **preguntar(Efecto)** per a cada efecte d'aquesta llista.

```
( Efectos \= [] ->
  forall(member(Efecto, Efectos), preguntar(Efecto))
```

- **Si no hi ha efectes:** Si no es troben efectes per al subsistema (és a dir, si la llista **Efectos** està buida), es mostra un missatge indicant que no hi ha efectes associats a aquest subsistema.

```
; format('No hay efectos para el subsistema: ~w~n',
[Subsistema])
)
)
),
```


Després de fer totes les preguntes, el sistema ha completat el diagnòstic i pot mostrar un missatge de conclusió, on l'usuari pot sol·licitar les causes possibles (**causas.**) i l'explicació del diagnòstic (**explicacion.**).

```
format('~n~`=t~60|~nDiagnostico completado.~nPuede solicitar las
posibles causas con "causas." y la explicacion con "explicacion."~n~`=t~
60|~n'),
```

Finalment, el predicat **esperar_comando/0** és cridat per permetre que l'usuari introdueixi un nou comandament, com "**causas.**" o "**explicacion.**" per obtenir més informació.

```
esperar_comando.
```

Preguntar per un efecte (preguntar)

Aquest predicat fa una pregunta a l'usuari sobre un efecte específic.

```
preguntar(Efecto) :-
```

Primer de tot, comprova que l'efecte no hagi estat preguntat abans.

```
not(preguntada(Efecto)),
```

Aquí es fa una pregunta a l'usuari sobre l'efecte.

```
format('¿Presenta el coche el siguiente efecto: ~w? (si/no) ',
[Efecto]),
```

Utilitzant el predicat **read/1** llegim la resposta que l'usuari escriu a la consola.

```
read(Respuesta),
```

Poden passar dues coses:

- La resposta és **si.**:
 - **assertz(respuesta(Efecto))**: Afegim el resultat afirmatiu a la base de dades.
 - **assertz(evidencia(Efecto))**: També l'afegim l'efecte a la llista d'evidències, ja que l'efecte s'ha confirmat.

```
( respuesta(Efecto) -> assertz(evidencia(Efecto)),
assertz(preguntada(Efecto))
```

- La resposta és **no**.:
 - **assertz(respuesta(no(Efecto)))** Registra la resposta negativa a la base de dades.

```
; respuesta(no(Efecto)) -> assertz(preguntada(Efecto))
```

Finalment amb **assertz(preguntada(Efecto))** marca la pregunta com a preguntada.

```
assertz(preguntada(Efecto)).
```

Verificar si un efecte ja ha estat preguntat (preguntada)

Aquest predicat verifica si un efecte ja ha sigut preguntat.

La primera línia del predicat verifica si existeix una resposta afirmativa per a l'efecte:

```
preguntada(Efecto) :-  
    respuesta(Efecto).
```

Això signific que si l'efecte té una resposta afirmativa (**respuesta(Efecto))** és certa), automàticament es considera que ja ha estat preguntat.

La segona línia del predicat comprova si hi ha una resposta negativa per a l'efecte:

```
preguntada(Efecto) :-  
    respuesta(no(Efecto)).
```

En aquest cas, si l'efecte té una resposta negativa (**respuesta(no(Efecto))** és certa), també es considera que ja ha estat preguntat.

No fer res quan ja s'ha preguntat l'efecte (preguntar(_))

Quan **preguntar(Efecto)** es crida i ja existeix **preguntada(Efecto)**, cap regla del predicat **preguntar(Efecto)** s'aplicarà. Per tant, el cas base s'executa quan l'efecte ja ha sigut preguntat, o sigui, **not(preguntada(Efecto))** ens hagi retornat **false**.

```
preguntar(_) :-  
    true.
```

Esperar un comandament de l'usuari (esperar_comando)

Aquest predicat es crida un cop finalitzat el diagnòstic i espera una entrada de l'usuari per tal d'executar l'acció corresponent segons el comandament introduït.

Primer de tot, mostra un missatge a l'usuari indicant les opcions disponibles:

- Escriure “**causas.**” per veure les causes probables.
 - Escriure “**explicacion.**” per obtenir l'explicació del diagnòstic.
 - Escriure “**nueva_sesion.**” per iniciar una nova sessió.
 - Escriure “**salir.**” per acabar el diagnòstic.
- ```
format('~nIngrese "causas." para ver las causas probables,
"explicacion." para ver la explicacion del diagnostico, o
"nueva_sesion." para iniciar una nueva sesion.~n'),
```

Amb el predicat **read**, es llegeix el que introdueix l'usuari, i en funció de l'entrada, es realitza una acció específica:

```
read(Comando),
```

- Si el comandament és **causas.**, es crida el predicat **causas** i després es torna a esperar un nou comandament.

```
(Comando == causas -> causas, esperar_comando
```

- Si el comandament és **explicacion.**, es crida el predicat **explicacion** i després es torna a esperar.

```
; Comando == explicacion -> explicacion, esperar_comando
```

- Si el comandament és **nueva\_sesion.**, es crida el predicat **nueva\_sesion** per reiniciar el procés.

```
; Comando == nueva_sesion -> nueva_sesion
```

- Si el comandament és **salir.**, es crida el predicat **salir**, que finalitza l'execució del programa.

```
; Comando == salir -> salir
```

- Si l'entrada no és reconeguda, es mostra un missatge d'error i es torna a esperar un comandament vàlid.

```
; format('Comando no reconocido. Por favor, intente de nuevo.\n'),
esperar_comando
).
```

## Causas del problema (causas)

Aquest predicat identifica les causes probables basades en les evidències observades i les mostra a l'usuari.

```
causas :-
```

### 1. Cerca de Causas:

S'utilitza el predicat **findall/3** per recopilar totes les causes relacionades amb els efectes observats.

Recordem que el predicat **findall** té la següent estructura: **findall(Causa, Condició, CausasDuplicadas)**.

**Causa** és l'element que busquem (les causes dels problemes):

```
findall(Causa,
```

**Condició** és el conjunt de regles que han de complir els elements per ser recollits:

Primerament, amb la condició **causa(\_, Causa)** busca totes les causes definides a la base de coneixement. Per exemple, si tenim aquest fet al domini:

```
causa(transmission, nivel_bajo_de_aceite_transmision).
causa(direccion, desgaste_de_neumaticos).
```

Aquesta condició retorna totes les causes una per una:

**'nivel\_bajo\_de\_aceite\_transmision'**

**'desgaste\_de\_neumaticos'**

```
(causa(_, Causa),
```

Després, verifiquem la relació entre una causa i els efectes esperats associats. Per exemple, si tenim aquest fet a la base de dades:

```
relacion(problemas_barra_de_direccion, [desviacion_hacia_un_lado,
coche_no_gira]).
relacion(fallo_en_los_bujes, [direccion_deficiente, ruido_al_girar]).
```

Aquesta condició busca quins efectes esperats (**EfectosEsperados**) estan associats a cada causa.

Aquí estem dient que:

Si la causa és **rotura\_de\_piezas**, els efectes esperats són:

- **desviacion\_hacia\_un\_lado**
- **coche\_no\_gira**

Si la causa és **fallo\_en\_los\_bujes**, els efectes esperats són:

- **direccion\_deficiente**
- **ruido\_al\_girar**

```
relacion(Causa, EfectosEsperados),
```

Seguidament, es verifica si un **efecte confirmat** (observat) està dins dels **efectes esperats** per a una causa.

Per exemple:

Si els efectes esperats són [**'desviacion\_hacia\_un\_lado'**, **'coche\_no\_gira'**]. I els efectes confirmats són [**'desviacion\_hacia\_un\_lado'**] (perquè l'usuari ha dit que **si.**) i [**'coche\_no\_gira'**]. Aquesta condició comprovarà que **'desviacion\_hacia\_un\_lado'** i **'coche\_no\_gira'** està present dins dels efectes esperats.

```
forall(member(Efecto, EfectosEsperados),
```

En cas contrari, si només tinguéssim l'efecte **'coche\_no\_gira'** o **'desviacion\_hacia\_un\_lado'** no seria possible la causa **'rotura\_de\_piezas'**.

Ara verifiquem si un efecte ha estat confirmat (resposta afirmativa). Això passa quan, en algun moment del programa, l'usuari ha respost **'si.'** a la pregunta:

**"¿Se presenta este problema/efecto: [nombre del efecto]?"**

**Important**, quan l'usuari respon **si.**, recorda que el sistema emmagatzema aquesta resposta com:

**assertz(respuesta(Efecto)).**

Així, el predicat **respuesta(Efecto)** esdevé **cert** només per als efectes confirmats per l'usuari.

Per exemple:

Si l'usuari ha respost **si.** per a l'efecte '**coche\_no\_gira**', llavors:

**respuesta('coche\_no\_gira').**

Aquesta condició filtra només els efectes que han estat confirmats com presents.

```
respuesta(Efecto)
),
```

## 2. Eliminació de duplicats:

**CausasDuplicadas** és la llista resultant amb totes les causes trobades (amb possibles duplicats).

```
CausasDuplicadas
),
```

A l'exemple anterior, si l'usuari a respost a **si.** a l'observació **coche\_no\_gira**, llavors **CausasDuplicadas** seria **[problemes\_barra\_direccion]**.

Un cop es té la llista **CausasDuplicadas**, s'utilitza **sort/2** per eliminar duplicats i ordenar la llista:

```
sort(CausasDuplicadas, CausasProbables),
```

Per exemple, si tenim **CausasDuplicadas = ['coche\_no\_gira', 'coche\_no\_gira', 'ruido\_al\_girar', 'ruido\_al\_girar', 'desviacion\_hacia\_un\_lado']**. El resultat de **CausasProbables** seria **['coche\_no\_gira', 'ruido\_al\_girar', 'desviacion\_hacia\_un\_lado']**.

### 3. Mostra de resultats:

Primerament, comencem comprovant si la llista de causes probables **no està buida** (**CausasProbables** `\= []`). Si la llista està buida, el codi no executarà les accions següents.

```
(CausasProbables \= [] ->
```

Amb **retractall/1** eliminem **qualsevol causa probable guardada prèviament** a la base de dades amb el predicat **causas\_probables()**.

```
retractall(causas_probables(_)),
```

Amb **assertz/1** afegim una nova entrada a la base de dades, col·locant-la al **final** de la memòria de fets.

```
assertz(causas_probables(CausasProbables)),
```

Amb **format/2** generem un missatge per a mostrar les causes probables a l'usuari:

```
format('~n~`=t~60|~nLas posibles causas son:~n~`=t~60|~n'),
```

Amb **forall/2** per a cada element **Causa** a la llista **CausasProbables**, es mostra un text amb **format/2**:

```
forall(member(Causa, CausasProbables),
 (format('~ - ~w~n', [Causa])
)),
```

Després mostrem un missatge indicant a l'usuari que pot escriure **explicacion.** per obtenir més detalls:

```
format('~nPuede solicitar la explicacion con "explicacion."~n')
```

En cas en que no hi hagi causes probables, es mostra un missatge indicant que no es poden concloure causes dels efectes observats:

```
; format('~nNo se puede concluir ninguna causa con los sintomas
proporcionados.~n')
).
```

## Explicació del diagnòstic (explicacion)

Si demanem **explicacion**, obtindrem l'explicació per part del diagnòstic, una cosa semblant com aquesta:

**Causa probable:** `fuita_al_sistema_hidràulic`

**Porque presenta estos efectos:** `[fuga_hidraulica, baja_presion]`

```
explicacion :-
```

Primerament, es consulta la base de dades dinàmica per obtenir la llista de causes probables prèviament generada amb el predicat **causas\_probables/1**.

```
causas_probables(CausasProbables),
```

Després, mostrem un missatge a l'usuari indicant que s'està generant l'explicació, amb una línia decorativa per fer-lo més clar i atractiu:

```
format('~n~`=t~60|~nExplicacion del diagnostico:~n~`=t~60|~n'),
```

Amb el predicat **forall/2** ens assegurem que es recorri cada causa a la llista **CausasProbables**. Per a cada causa, es generarà una explicació detallada:

```
forall(member(Causa, CausasProbables), ...)
```

1. Mostrem el nom de la causa actual amb el predicat **format/2**:

```
(format('Causa probable: ~w~n', [Causa]),
```

2. Amb **relacion/2**, busquem quins efectes s'esperen que estiguin associats amb la causa actual.

```
relacion(Causa, EfectosEsperados),
```

3. Amb **findall/3** creem una llista de tots els efectes que compleixen dues condicions:

- 3.1. **member(Efecto, EfectosEsperados)**: L'efecte ha de ser un dels efectes esperats associats a la causa.

```
findall(Efecto, (member(Efecto, EfectosEsperados),
```

- 3.2. **evidencia(Efecto)**: L'efecte ha de ser una evidència realment observada.

```
evidencia(Efecto)),
```



El resultat es guarda a **Evidencias**, que conté els efectes esperats confirmats per les observacions.

Si hi ha **evidències observades**:

Es mostren els efectes confirmats que donen suport a aquesta causa:

```
(Evidencias \= [] ->
 format('Porque presenta estos efectos: ~w~n',
[Evidencias])
```

Si **no** hi ha **evidències**:

```
; format('No se encontraron efectos asociados para la
causa: ~w~n', [Causa])
)
)).
```

Si no hi ha coincidències entre els efectes esperats i les evidències observades, es mostra un missatge indicant que aquesta causa no està recolzada per cap evidència.

## Per iniciar una nova sessió (nueva\_sesion)

Permet iniciar una nova sessió de diagnòstic.

```
nueva_sesion :-
```

El predicat comença amb **format/2** per mostrar la pregunta a l'usuari preguntant-li si vol iniciar una nova sessió o no:

```
format('~n¿Desea iniciar una nueva sesion? (si/no) '),
```

Després, s'utilitza el predicat **read/1** per llegir la resposta de l'usuari. El valor llegit es guarda en la variable **Respuesta**:

```
read(Respuesta),
```

Si **Respuesta == si**, és a dir, l'usuari vol iniciar una nova sessió, s'executa el predicat **main/0**:

```
(Respuesta == si -> main
```

Si la resposta és diferent de **si**, en aquest cas, si l'usuari ha respost **no**, s'executa el predicat **salir/0**:

```
; salir
).
```

### Predicat per sortir (salir)

El predicat **salir/0** permet sortir del programa quan l'usuari decideix finalitzar la sessió, mostrant un missatge de confirmació i aturant l'execució del programa.

El predicat comença amb **format/2** per mostrar a l'usuari un missatge indicant que la sessió s'ha acabat:

```
format('~nSesion terminada.~n'),
```

Després fem servir el **cut (!)** per evitar retrocedir en el flux de control.

```
!,
```

Finalment, el predicat crida **halt/0**. Això fa que el programa es tanqui completament després de mostrar el missatge de terminació de la sessió.

```
halt.
```

# MANUAL D'USUARI PER AL SISTEMA DE DIAGNÒSTIC D'AVERIES

Aquest manual descriu com utilitzar el sistema per diagnosticar possibles averies basades en els efectes observats en diferents subsistemes del domini.

El sistema està basat en Prolog i permet realitzar un diagnòstic interactiu, o bé carregant efectes ja observats o bé que el sistema mateix et faci preguntes per detectar i explicar les possibles causes.

Abans que res, descarrega la carpeta anomenada **Paz** que es troba al nostre repositori de Github (podeu trobar l'enllaç a **Observacions**).

Una vegada descarregada, copia l'enllaç de la ruta de la carpeta on es troba perquè haurem de fer un canvi en el codi de **SolucioOscarLucia.pl**. A la línia 160 i 186 en vegada de **'C:/Users/usuari/OneDrive/Escritorio/Practica/Paz/'** posa la teva ruta.

Després d'haver fet això, obrim **SWI-PROLOG**, ves a l'apartat superior a l'esquerra i clica **File**, després **Consult...** i importa **SolucioOscarLucia.pl**.

## 1. Iniciar el sistema

### 1.1 Iniciar el diagnòstic

Executa el predicat **main/0** per iniciar el diagnòstic:

```
?- main.

=====
Bienvenido al sistema de diagnostico de averias.
=====
Cargue un sistema (ponga el nombre del archivo sin extension .pl):
```

El sistema comença sol·licitant a l'usuari que carregui un fitxer de coneixement, o sigui, el domini. En aquest cas tenim dues opcions: **coche.pl** o **cocina.pl**. Suposem que importem la base de coneixement de **coche.pl**.

```
Cargue un sistema (ponga el nombre del archivo sin extension .pl): coche.
Archivo coche.pl cargado correctamente.
¿Desea cargar hechos observados desde un archivo? (si/no) |:
```

Després de carregar el fitxer de base de coneixement, el sistema et preguntarà si vols carregar fets observats (és a dir, les averies que ja has notat). Si respon **"si."**, es carregaran els fets observats des d'un fitxer. Si respon **"no."** el sistema et farà preguntes per tal d'identificar les possibles causes a partir dels efectes que has observat.

## 1.2 Carregar fets observats

Si vols carregar fets observats has de respondre **"si."**:

```
¿Desea cargar hechos observados desde un archivo? (si/no) |: si.
Ingrese el nombre del archivo de hechos observados (sin extension .pl): |:
```

En aquest cas tens dues opcions, o bé carregar **hechos\_coche**. O bé **hechos\_cocina..** Suposem que hem carregat **hechos\_coche.**:

```
Ingrese el nombre del archivo de hechos observados (sin extension .pl): |: hechos_coche.
Hechos observados cargados desde hechos_coche.pl
Diagnostico completado.
Puede solicitar las posibles causas con "causas." y la explicacion con "explicacion."

Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion
del diagnostico, o "nueva_sesion." para iniciar una nueva sesion.
|: █
```

Ara pots demanar al sistema les possibles causes amb **causas.** o que et doni les explicacions **explicacion.** de les causes.

```
Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion
del diagnostico, o "nueva_sesion." para iniciar una nueva sesion.
|: causas.

=====
Las posibles causas son:
=====
- desgaste_de_neumaticos
- deterioro_de_los_silentblocks

Puede solicitar la explicacion con "explicacion."

Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion
del diagnostico, o "nueva_sesion." para iniciar una nueva sesion.
|: explicacion.

=====
Explicacion del diagnostico:
=====
Causa probable: desgaste_de_neumaticos
Porque presenta estos efectos: [vibraciones_volante]
Causa probable: deterioro_de_los_silentblocks
Porque presenta estos efectos: [vibraciones_volante]
```

Si ja no vols veure ni les causes ni les explicacions, pot escriure **nueva\_sesion** per començar de nou el diagnòstic o sortir del sistema.

```
Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion
del diagnostico, o "nueva_sesion." para iniciar una nueva sesion.
|: nueva_sesion.

¿Desea iniciar una nueva sesion? (si/no) |:
```

## 2. Diagnòstic

Suposem que no volem carregar fets observats, per tant, posem **no.**:

```

¿Desea cargar hechos observados desde un archivo? (si/no) |: no.
=====
Preguntas para el subsistema: direccion
=====
¿Se presenta este problema/efecto: vibraciones_volante? (si/no) |: ■

```

Ara hem d'escriure **si.** o **no.** depenent de si es presenta o no l'efecte que ens pregunta el sistema. Després d'haver respost totes les preguntes, ja podrem veure les possibles causes i explicacions d'aquestes per part del diagnòstic:

```

=====
Diagnostico completado.
Puede solicitar las posibles causas con "causas." y la explicacion con "explicacion."
=====
Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion
del diagnostico, o "nueva_sesion." para iniciar una nueva sesion.
|:

```

## RESULTATS

En aquesta secció documentarem els casos concrets en què hem provat el sistema i els resultats que han donat. Aquests casos mostren com el sistema respon a diferents situacions, comprovant si funciona correctament.

Suposem que no carreguem un fitxer que conté els fets ja observats, sinó que el sistema ens preguntí sobre els efectes que hem observat del domini **coche.pl**.

```
?- main.

=====
Bienvenido al sistema de diagnostico de averias.
=====
Cargue un sistema (ponga el nombre del archivo sin extension .pl): coche.
Archivo coche.pl cargado correctamente.
¿Desea cargar hechos observados desde un archivo? (si/no) |: no.
```

Una vegada fet això, el sistema ens preguntarà sobre les observacions que hem fet en cada subsistema del cotxe:

```
=====
Preguntas para el subsistema: direccion
=====

¿Se presenta este problema/efecto: vibraciones_volante? (si/no) |:
```

Suposem que hem observat els següents efectes:

```
=====
Preguntas para el subsistema: direccion
=====

¿Se presenta este problema/efecto: vibraciones_volante? (si/no) |: si.
¿Se presenta este problema/efecto: ruido_al_girar? (si/no) |: no.
¿Se presenta este problema/efecto: direccion_dura? (si/no) |: no.
¿Se presenta este problema/efecto: coche_no_gira? (si/no) |: no.
¿Se presenta este problema/efecto: desviacion_hacia_un_lado? (si/no) |: no.

=====
Preguntas para el subsistema: transmission
=====

¿Se presenta este problema/efecto: dificultad_para_cambiar_marchas? (si/no) |: si.
¿Se presenta este problema/efecto: perdida_de_potencia? (si/no) |: no.
¿Se presenta este problema/efecto: ruidos_extranos_en_la_caja? (si/no) |: no.
¿Se presenta este problema/efecto: vibraciones_al_conducir? (si/no) |: no.
¿Se presenta este problema/efecto: sobrecalentamiento? (si/no) |: si.
```

```

=====
Preguntas para el subsistema: encesa
=====

¿Se presenta este problema/efecto: el_motor_no_arranca? (si/no) |: no.
¿Se presenta este problema/efecto: ralenti_irregular? (si/no) |: no.
¿Se presenta este problema/efecto: consumo_excesivo_combustible? (si/no) |: no.
¿Se presenta este problema/efecto: detonaciones? (si/no) |: no.
¿Se presenta este problema/efecto: chispa_debil? (si/no) |: si.
=====
Preguntas para el subsistema: sistema_electrico
=====

¿Se presenta este problema/efecto: luces_no_funcionan? (si/no) |: no.
¿Se presenta este problema/efecto: bateria_descargada? (si/no) |: no.
¿Se presenta este problema/efecto: alternador_fallido? (si/no) |: no.
¿Se presenta este problema/efecto: cortocircuitos? (si/no) |: no.
¿Se presenta este problema/efecto: fusibles_fundidos? (si/no) |: no.
¿Se presenta este problema/efecto: sistema_no_responde? (si/no) |: no.
¿Se presenta este problema/efecto: sobrecalentamiento_cables? (si/no) |: no.

=====
Preguntas para el subsistema: sistema_confort_interior
=====

¿Se presenta este problema/efecto: fallos_en_el_sistema_de_climatizacion? (si/no.
¿Se presenta este problema/efecto: problemas_en_el_sistema_de_audio? (si/no) |no.
¿Se presenta este problema/efecto: problema_para_subir_y_bajar_las_ventanas? (no.
¿Se presenta este problema/efecto: no_se_ajustan_los_asientos? (si/no) |: no.
¿Se presenta este problema/efecto: no_funciona_el_cierre_centralizado? (si/no)si.
¿Se presenta este problema/efecto: no_funciona_las_luces_interiores? (si/no) |no.
=====
Diagnostico completado.
Puede solicitar las posibles causas con "causas." y la explicacion con "explicacion.
"
=====

```

A partir d'aquests efectes, les possibles causes que ens hauria de donar el sistema serien les següents:

Efecte **vibraciones\_volante**:

**desgaste\_de\_neumaticos** (línia 56), **deterioro\_de\_los\_silentblocks** (58).

Efecte **dificultad\_para\_cambiar\_marchas**:

**desgaste\_sincronizadores** (87)

Efecte **sobrecalentamiento**:

**filtro\_obstruido** (89).

Efecte **dificultad\_para\_cambiar\_marchas** i **sobrecalentamiento**:

**nivel\_bajo\_de\_aceite\_transmision (86)**

Efecte **chispa\_debil**:

**No es pot derivar cap causa a partir d'aquest únic efecte**

Efecte **no\_funciona\_el\_cierre\_centralizado**:

**problema\_con\_el\_solenoid (196).**

En efecte, si volguéssim veure les causes amb **causa.** ens surt una llista amb aquestes possibles causes:

```
Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion
del diagnostico, o "nueva_sesion." para iniciar una nueva sesion.
|: causas.
```

```
=====
Las posibles causas son:
=====
- desgaste_de_neumaticos
- desgaste_sincronizadores
- deterioro_de_los_silentblocks
- filtro_obstruido
- nivel_bajo_de_aceite_transmision
- problema_con_el_solenoid
```

Puede solicitar la explicacion con "explicacion."

Si volguéssim veure les explicacions que dona el sistema de com ha arribat a aquestes possibles causes fem **explicacion.:**

```
Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion
del diagnostico, o "nueva_sesion." para iniciar una nueva sesion.
|: explicacion.
```

```
=====
Explicacion del diagnostico:
=====
Causa probable: desgaste_de_neumaticos
Porque presenta estos efectos: [vibraciones_volante]
Causa probable: desgaste_sincronizadores
Porque presenta estos efectos: [dificultad_para_cambiar_marchas]
Causa probable: deterioro_de_los_silentblocks
Porque presenta estos efectos: [vibraciones_volante]
Causa probable: filtro_obstruido
Porque presenta estos efectos: [sobrecalentamiento]
Causa probable: nivel_bajo_de_aceite_transmision
Porque presenta estos efectos: [dificultad_para_cambiar_marchas,sobrecalentamiento]
Causa probable: problema_con_el_solenoid
Porque presenta estos efectos: [no_funciona_el_cierre_centralizado]
```

També ens hauria de sortir el mateix si tinguéssim aquests efectes al fitxer **hechos\_coche.pl**. En efecte, ens surt el mateix:



```

Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion
del diagnostico, o "nueva_sesion." para iniciar una nueva sesion.
|: causas.

=====
Las posibles causas son:
=====
- desgaste_de_neumaticos
- desgaste_sincronizadores
- deterioro_de_los_silentblocks
- filtro_obstruido
- nivel_bajo_de_aceite_transmision
- problema_con_el_solenoide

Puede solicitar la explicacion con "explicacion."

Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion
del diagnostico, o "nueva_sesion." para iniciar una nueva sesion.
|: explicacion.

=====
Explicacion del diagnostico:
=====
Causa probable: desgaste_de_neumaticos
Porque presenta estos efectos: [vibraciones_volante]
Causa probable: desgaste_sincronizadores
Porque presenta estos efectos: [dificultad_para_cambiar_marchas]
Causa probable: deterioro_de_los_silentblocks
Porque presenta estos efectos: [vibraciones_volante]
Causa probable: filtro_obstruido
Porque presenta estos efectos: [sobrecalentamiento]
Causa probable: nivel_bajo_de_aceite_transmision
Porque presenta estos efectos: [dificultad_para_cambiar_marchas,sobrecalentamiento]
Causa probable: problema_con_el_solenoide
Porque presenta estos efectos: [no_funciona_el_cierre_centralizado]

```

Perquè el sistema no només funcioni amb el domini del **coche.pl**, ara farem el mateix amb el domini de la cuina (**cocina.pl** i **hechos\_cocina.pl**). Per tant, escrivim **nueva\_sesion** per iniciar un nou diagnòstic i posem **si.**:

```

Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion
del diagnostico, o "nueva_sesion." para iniciar una nueva sesion.
|: nueva_sesion.

¿Desea iniciar una nueva sesion? (si/no) |: si.

=====
Bienvenido al sistema de diagnostico de averias.
=====
Cargue un sistema (ponga el nombre del archivo sin extension .pl): |:

```

Carreguem la base de coneixement de la cuina escrivint **cocina.**:

```

=====
Bienvenido al sistema de diagnostico de averias.
=====
Cargue un sistema (ponga el nombre del archivo sin extension .pl): |: cocina.

Archivo cocina.pl cargado correctamente.
¿Desea cargar hechos observados desde un archivo? (si/no) |: ■

```

Suposem que no volem carregar fets observats, per tant, posem **no.**:

```

Archivo cocina.pl cargado correctamente.
¿Desea cargar hechos observados desde un archivo? (si/no) |: no.

=====
Preguntas para el subsistema: horno
=====

```

Ara el sistema ens farà preguntes sobre els efectes que hem observat, suposem que hem vist les següents observacions:

```
=====
Preguntas para el subsistema: horno
=====

¿Se presenta este problema/efecto: no_calienta? (si/no) |: si.
¿Se presenta este problema/efecto: calienta_demasiado? (si/no) |: no.
¿Se presenta este problema/efecto: hace_ruido? (si/no) |: no.
=====
Preguntas para el subsistema: fogones
=====

¿Se presenta este problema/efecto: no_encienden? (si/no) |: no.
¿Se presenta este problema/efecto: llama_baja? (si/no) |: no.
¿Se presenta este problema/efecto: fuga_de_gas? (si/no) |: si.
=====
Preguntas para el subsistema: refrigerador
=====

¿Se presenta este problema/efecto: no_enfria? (si/no) |: no.
¿Se presenta este problema/efecto: fuga_de_agua? (si/no) |: no.
=====
Diagnostico completado.
Puede solicitar las posibles causas con "causas." y la explicacion con "explicacion."
=====
```

Segons aquests efectes, les causes probables haurien de ser (pots anar a **cocina.pl**):

Efecte **no\_calienta**:

**resistencia\_quemada** (línia 29)

Efecte **fuga\_de-gas**:

**manguera\_rota** (47)

En efecte, si volem veure la llista de causes amb **causas.** ens surt aquestes dues causes probables:

```
Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion
del diagnostico, o "nueva_sesion." para iniciar una nueva sesion.
|: causas.
=====
Las posibles causas son:
=====
- manguera_rota
- resistencia_quemada

Puede solicitar la explicacion con "explicacion."
```

Sollicitem la explicació amb **explicacion.:**

Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion del diagnostico, o "nueva\_sesion." para iniciar una nueva sesion.  
|: explicacion.

```
=====
Explicacion del diagnostico:
=====
Causa probable: manguera_rota
Porque presenta estos efectos: [fuga_de_gas]
Causa probable: resistencia_quemada
Porque presenta estos efectos: [no_calienta]
```

Ara fiquem aquests efectes observats al nostre fitxer **hechos\_cocina.pl**:

```
respuesta(no_calienta).
respuesta(no(no_encienden)).
respuesta(no(hace_ruido)).
respuesta(fuga_de_gas).
respuesta(no(no_enfria)).
respuesta(no(fuga_de_agua)).
respuesta(no(llama_baja)).
respuesta(no(calienta_demasiado)).
respuesta(no(hace_ruido)).
```

Ens hauria de sortir el mateix, fem **nueva\_sesion. -> si. -> cocina. -> si. -> hechos\_cocina.:**

Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion del diagnostico, o "nueva\_sesion." para iniciar una nueva sesion.  
|: nueva\_sesion.

```
¿Desea iniciar una nueva sesion? (si/no) |: si.
=====
Bienvenido al sistema de diagnostico de averias.
=====
Cargue un sistema (ponga el nombre del archivo sin extension .pl): |: cocina.
Archivo cocina.pl cargado correctamente.
¿Desea cargar hechos observados desde un archivo? (si/no) |: si.
Ingrese el nombre del archivo de hechos observados (sin extension .pl): |: hechos_cocina.
Hechos observados cargados desde hechos_cocina.pl
Diagnostico completado.
Puede solicitar las posibles causas con "causas." y la explicacion con "explicacion."
```

En efecte, ens surt el mateix si demanem **causas. i explicacion.:**

Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion del diagnostico, o "nueva\_sesion." para iniciar una nueva sesion.  
|: causas.

```
=====
Las posibles causas son:
=====
- manguera_rota
- resistencia_quemada
```

Puede solicitar la explicacion con "explicacion."

Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion del diagnostico, o "nueva\_sesion." para iniciar una nueva sesion.  
|: explicacion.

```
=====
Explicacion del diagnostico:
=====
Causa probable: manguera_rota
Porque presenta estos efectos: [fuga_de_gas]
Causa probable: resistencia_quemada
Porque presenta estos efectos: [no_calienta]
```

Ingrese "causas." para ver las causas probables, "explicacion." para ver la explicacion del diagnostico, o "nueva\_sesion." para iniciar una nueva sesion.  
|: ■

D'aquesta manera, hem comprovat que el sistema no només funciona amb el domini dels cotxes, sinó que també funciona amb el domini de les cuines.

## CONCLUSIONS

El projecte ens ha permès posar en pràctica els conceptes de programació lògica per crear un sistema capaç de diagnosticar avaries de manera interactiva i precisa. Hem aconseguit desenvolupar una eina que funciona tant per a automòbils com per a cuines, complint els requisits del nivell d'ambició màxim establert a la pràctica.

Durant el procés, hem après a estructurar coneixement complex mitjançant regles i relacions causals, i a gestionar un diàleg interactiu amb l'usuari que facilita l'obtenció de diagnòstics útils. Un dels punts més importants ha estat garantir la flexibilitat del sistema, que es pot ampliar fàcilment a altres dominis.

També volem donar una especial importància al document ***La Diagnosi Com a Tasca Cognitiva.pdf***, ja que va ser la clau que ens va permetre assolir aquest sistema. No sabíem com començar, però gràcies al contingut del fitxer vam entendre com havia de ser el sistema en qüestió, tant la interacció amb l'usuari com l'explicació per part del de com havia arribat a aquelles causes probables.

En general, creiem que el sistema és funcional i ofereix una base sòlida per continuar explorant aplicacions similars en altres àmbits. Tot i les dificultats trobades inicialment, com la gestió de fitxers i l'organització del codi, estem satisfets amb els resultats obtinguts.

## OBSERVACIONS

Aquí us deixem el nostre repositori de Github on podeu trobar, entre altres coses, el **SolucioOscarLucia.pl**, **coche.pl**, **cocina.pl**, **hechos\_coche.pl**, **hechos\_cocina.pl**:  
<https://github.com/bboscar248/PROLOG>

## BIBLIOGRAFIA

alziraimport. (2024, March 18). *¿Problemas con las ventanillas del coche? Aprende a solucionarlos y a mantenerlas en buen estado - alziraimport.* Alziraimport. <https://alziraimport.es/elevallunas-no-suben-ni-baja/#ContinuarLeyendo>

Aprende Institute. (2022, September 16). *¿Por qué mi refrigerador no enfría? | Aprende Institute.* Aprende Institute. <https://aprende.com/blog/oficios/repuracion-electronica/fallas-de-un-refrigerador-que-no-enfria/>

Autoingeniería. (2024). *Los 5 signos más comunes de que el sistema eléctrico de tu coche está fallando | Auto Ingeniería.* Autoingeniería.com. <https://www.autoingenieria.com/los-5-signos-mas-comunes-de-que-el-sistema-electrico-de-tu-coche-esta-fallando.html>

BBC. (2016, September 20). *Por qué pierden agua los refrigeradores y qué puedes hacer para evitarlo - BBC News Mundo.* BBC News Mundo. <https://www.bbc.com/mundo/noticias-37418224>

Binary Electronic. (2023, January 13). *Reparación centralitas sistemas de confort - BinaryElectronic.* Binary Electronic - Taller Electrónica Del Automóvil. <https://binaryelectronic.com/ reparar-centralitas-sistema-confort/>

Binary Electronic . (2023, August 22). *Fallos en el Sistema de Climatización de un Coche: Causas, Diagnóstico y Soluciones - Binary Electronic Solution.* Binary Electronic - Taller Electrónica Del Automóvil. [https://binaryelectronic.com/fallos-en-el-sistema-de-climatizacion-de-un-coche-causas-diagnostico-y-soluciones/?utm\\_source](https://binaryelectronic.com/fallos-en-el-sistema-de-climatizacion-de-un-coche-causas-diagnostico-y-soluciones/?utm_source)

Blanco, J. (2023, July 13). *Solución fácil: Luces interiores que no se encienden al abrir la puerta.* Electrium.es; Electrium. <https://electrium.es/iluminacion/encienden-luces-interiores-abrir-puerta/>

Carglass. (2024, October 28). *Los problemas más comunes en el sistema de transmisión. Blog Del Experto En Reparación Y Sustitución de Lunas | Carglass®.* <https://www.carglass.es/blog/coche-a-punto/problemas-comunes-sistema-transmision/>

CASTILLO, D. (2023, October 11). *No funciona el cierre centralizado de una puerta: causas y soluciones* | Reparaciones10.com. Reparaciones10.com. <https://reparaciones10.com/no-funciona-el-cierre-centralizado-de-una-puerta-causas-y-soluciones/>

Confortauto. (2020, November 4). *Fallos eléctricos de coche más comunes: causa y solución*. Blog Confortauto. <https://www.confortauto.com/blog/confortauto/cinco-fallos-electricos-que-suelen-sufrir-los-coches/?srsltid=AfmBOorEltG1z5ip-oqxtWcSROrRaUx3mRqGfZFuQwR4gAD9861d6rMz>

Den. (2020, May 28). *Señales de que un sistema de encendido se encuentra defectuoso — Cómo inspeccionar el sistema de encendido*. Autodoc.es; Autodoc Club. <https://club.autodoc.es/magazin/senales-de-que-un-sistema-de-encendido-se-encuentra-defectuoso-como-inspeccionar-el-sistema-de-encendido>

Experto Mi Electro. (2021, January 28). *¿Ruido en la nevera? Causas y soluciones. [GUÍA COMPLETA]*. Mi Electro News. <https://www.mielectro.es/blog/consejos-trucos/ruido-nevera-causas-posibles-soluciones/>

Inteligentes. (n.d.). *Resolución de problemas comunes de audio para automóviles*. Inteligentes.online. Retrieved December 27, 2024, from <https://www.inteligentes.online/Car-Audio-y-Electr%C3%B3nica/Altavoces-del-coche-y-Subwoofers/Resoluci%C3%B3n-de-problemas-comunes-de-audio-para-autom%C3%B3viles.html>

Isaac. (2024, October 21). *Averías en el sistema de dirección de tu coche*. Actualidad Motor. <https://www.actualidadmotor.com/averias-sistema-direccion-coche/>

Lagosat. (2023, December 28). *Identifica los problemas más comunes de un horno averiado*. LAGOSAT. <https://lagosat.com/identifica-problemas-comunes-horno-averiado/>

makeadito. (2023, October 12). *Descubre cómo solucionar el problema de la luz interior del coche que no enciende - Makeadito*. Makeadito. <https://makeadito.es/luz-interior-coche-no-enciende/>



Páxinas Galegas. (2023, November 7). *Cómo detectar y solucionar los problemas eléctricos y electrónicos en su coche*. Dieselinyeccionmilladoiro.es. [https://www.dieselinyeccionmilladoiro.es/como-detectar-y-solucionar-los-problemas-electricos-y-electronicos-en-su-coche\\_fb156112.html](https://www.dieselinyeccionmilladoiro.es/como-detectar-y-solucionar-los-problemas-electricos-y-electronicos-en-su-coche_fb156112.html)

Pemebla. (2023, April 17). *¿Cómo detectar las averías del sistema de encendido del coche?* - Pemebla. Pemebla. <https://pemebla.com/blog/averias-sistema-encendido-coche-pemebla/>

RACE. (2023, February 22). *Resetear el cierre centralizado: ¿cómo se hace y cuándo debes hacerlo?* RACE. <https://www.race.es/resetear-cierre-centralizado>

Swi-prolog. (2021). *SWI-Prolog -- Command line options*. Swi-Prolog.org. [https://www.swi-prolog.org/pldoc/man?section=cmdline&utm\\_source](https://www.swi-prolog.org/pldoc/man?section=cmdline&utm_source)

Swi-prolog. (2023). *SWI-Prolog -- Getting started quickly*. Swi-Prolog.org. [https://www.swi-prolog.org/pldoc/man?section=quickstart&utm\\_source](https://www.swi-prolog.org/pldoc/man?section=quickstart&utm_source)

Swi-prolog. (2024). *SWI-Prolog -- Frequently used editor commands*. Swi-Prolog.org. [https://www.swi-prolog.org/pldoc/man?section=pceemacscommands&utm\\_source](https://www.swi-prolog.org/pldoc/man?section=pceemacscommands&utm_source)

Valentino. (2024, November 10). *consejos expertos y pasos sencillos*. Unión Por El Motor Racing. <https://www.upmracing.es/como-arreglar-la-palanca-del-asiento-del-coche/>

Washerhouse. (2025). *El fogó no s'encén: trobem les causes del fracàs en diferents models i la reparem amb les nostres pròpies mans*. Washerhouse.com. <https://top.washerhouse.com/ca/9/dlja-kuhni/kuhonnaja-plita/varochnaja-panel-ne-vkljuchaetsja/>