

# REPRESENTACIÓN DEL CONOCIMIENTO

## Tarea 3

Angel García Báez  
Alumno de la Maestría en Inteligencia Artificial

**IIIA** Instituto de Investigaciones en Inteligencia Artificial  
Universidad Veracruzana  
*Campus Sur, Calle Paseo Lote II, Sección 2a, No 112*  
*Nuevo Xalapa, Xalapa, Ver., México 91097*

ZS24019400@estudiantes.uv.mx

18 de junio de 2025

## 1. Representación en Prolog del universo del discurso y las relaciones entre objetos

Con base en el lineamiento estandarizado para la vigilancia epidemiológica y por laboratorio de COVID-19:

Represente en Prolog el universo de discurso y la relaciones entre objetos pertinentes para las páginas 11-17. (50/100).

Para la obtención del conocimiento primario fue necesario leer y releer varias veces la normativa que indica el manual hasta lograr concretar los puntos claves del mismo. En el manual se exponen normativas sobre la identificación de casos sospechosos y casos confirmados junto con normativas sobre las medidas de prevención, países de riesgo, comportamientos como viajar o tener contacto con personas sospechosas o confirmadas, así como rigurosas medidas de precaución, prevención y avisar en caso de detectar casos confirmados, como se debe proceder en dichos casos.

Como lo que se busca principalmente es la identificación de casos mediante sus antecedentes, la base del conocimiento se enfoco más en esa parte pero incluyendo también en buena medida un poco de lo relacionado con los médicos, las precauciones y las acciones a realizar.

Dentro del universo del discurso se tienen a las personas (que puede ser cualquier individuo que reside en México), donde la persona puede pasar a ser paciente si se detecta que es un caso sospechoso o un caso confirmado.

A continuación se muestra la base de datos del conocimiento que trata de fungir como una especie de sistema experto para realizar consultas sobre algunos sujetos declarados en el programa:

```
% Declaración de dinámicos
:- dynamic casoSospechoso/1.
:- dynamic casoConfirmado/1.
% -----
% Personal de salud
% -----
%% Tipo %%
personal_salud(X) :- enfermera(X).
personal_salud(X) :- epidemiologo(X).
%% Subtipo %%
enfermera(enfermera1).
epidemiologo(epidemiolo1).
% -----
```

```

% Personas
% -----
persona(juan).
persona(maria).
persona(pedro).
% -----
% Sintomatología
% -----
presenta_sintoma(juan, fiebre).
presenta_sintoma(juan, tos).
presenta_sintoma(juan, dificultad_respiratoria).
tiene_enfermedad_respiratoria(juan).
% -----
% Viajes y contactos
% -----
% Viajes
viajo_a(juan, china).
% Contacto
contacto_directo(juan, maria).
contacto_con(A, B) :-
contacto_directo(A, B);
contacto_directo(B, A).
% -----
% Resultado laboratorio
% -----
resultado_laboratorio(juan, positivo).
% -----
% Uso de mascarilla
% -----
usa_mascarilla(juan).
% -----
% Niveles de atención
% -----
nivel_atencion(juan, primer_nivel).
nivel_atencion(maria, tercer_nivel).
% -----
% Países de riesgo
% -----
pais_riesgo(china).
pais_riesgo(italia).

```

```

pais_riesgo(iran).
pais_riesgo(japon).
pais_riesgo(hong_kong).
pais_riesgo(corea_del_sur).
pais_riesgo(singapur).
% -----
% Precauciones estándar
% -----
precaucion_estandar(higiene_de_manos).
precaucion_estandar(uso_de_guantes).
precaucion_estandar(bata_impermeable).
precaucion_estandar(uso_mascarilla).
precaucion_estandar(uso_contenedores).
% -----
% Precauciones por gotas
% -----
precaucion_gotas(uso_equipo_desechable).
precaucion_gotas(mascarilla_quirurgica).
precaucion_gotas(distancia_metro).
% -----
% Precauciones por aerosoles
% -----
precaucion_aerosoles(respirador_n95).
% -----
% Aplicación de precauciones por personal
% -----
aplica(enfermera1, higiene_de_manos).
aplica(enfermera1, uso_de_guantes).
aplica(enfermera1, bata_impermeable).
aplica(enfermera1, proteccion_facial).
aplica(enfermera1, mascarilla_quirurgica).
aplica(enfermera1, respirador_n95).
% Verificar precaucion aplicada
precaucion_aplicada_a(Precaucion, Personal) :-
aplica(Personal, Precaucion).
% -----
% Toma de muestras
% -----
tipo_muestra(exudado_nasofaringeo).
tipo_muestra(exudado_faringeo).

```

```

tipo_muestra(lavado_bronquioalveolar).
tipo_muestra(biopsia_pulmonar).
toma_muestra(enfermera1, juan, exudado_nasofaringeo).
toma_muestra(enfermera1, juan, exudado_faringeo).
% -----
% Aislamiento aplicado
% -----
aislamiento_recomendado(juan, hospitalario_individual).
aislamiento_recomendado(maria, domicilio).
% -----
% Definición de caso sospechoso
% -----
% un confirmado ya no debe registrarse como sospechoso
esCasoSospechoso(Persona) :-
    casoConfirmado(Persona),
    format("Error: ~w ya es un caso confirmado, no puede registrarse como sospechoso.~n",
    esCasoSospechoso(Persona) :-
    casoSospechoso(Persona),
    write(Persona), write(" ya está registrado como caso sospechoso."), nl, !.
% Variante 1: por viaje a país de riesgo
esCasoSospechoso(Persona) :-
    tiene_enfermedad_respiratoria(Persona),
    viaje_a(Persona, Pais),
    pais_riesgo(Pais),
    format("~w cumple los criterios de caso sospechoso por viaje reciente a país de riesgo",
    asserta(casoSospechoso(Persona)), !.
% Variante 2: por contacto con caso confirmado
esCasoSospechoso(Persona) :-
    tiene_enfermedad_respiratoria(Persona),
    contacto_con(Persona, Otro),
    esCasoConfirmado(Otro),
    format("~w cumple los criterios de caso sospechoso por contacto reciente con caso confirmado",
    asserta(casoSospechoso(Persona)), !.
% Si no cumple definición
esCasoSospechoso(Persona) :-
    write(Persona), write(" no cumple definición de caso sospechoso."), nl,
    fail.
% -----
% Definición de caso confirmado
% -----

```

```

esCasoConfirmado(Persona) :-
    casoConfirmado(Persona),
    write(Persona), write(" ya está registrado como caso confirmado."), nl, !.
esCasoConfirmado(Persona) :-
    esCasoSospechoso(Persona),
    resultado_laboratorio(Persona, positivo),
    format("~w es un caso confirmado porque su prueba de laboratorio resultó positiva.~n",
    asserta(casoConfirmado(Persona)), !.
esCasoConfirmado(Persona) :-
    write(Persona), write(" no cumple definición de caso confirmado."), nl,
    fail.
%% precauciones %%
% Verificador general para cualquier categoría de precauciones
cumple_todas(Personal, Categoria, PrecaucionList) :-
    forall(member(Precaucion, PrecaucionList),
    aplica(Personal, Precaucion)).
% Definición para cada tipo de precaución
cumple_precauciones_estandar(Personal) :-
    findall(Prec, precaucion_estandar(Prec), Lista),
    ( cumple_todas(Personal, estandar, Lista)
    -> write(Personal), write(" cumple todas las precauciones estándar."), nl
    ; write(Personal), write(" NO cumple todas las precauciones estándar."), nl,
    mostrar_faltantes(Personal, estandar)
    ).
cumple_precauciones_gotas(Personal) :-
    findall(Prec, precaucion_gotas(Prec), Lista),
    ( cumple_todas(Personal, gotas, Lista)
    -> write(Personal), write(" cumple todas las precauciones por gotas."), nl
    ; write(Personal), write(" NO cumple todas las precauciones por gotas."), nl,
    mostrar_faltantes(Personal, gotas)
    ).
cumple_precauciones_aerosoles(Personal) :-
    findall(Prec, precaucion_aerosoles(Prec), Lista),
    ( cumple_todas(Personal, aerosoles, Lista)
    -> write(Personal), write(" cumple todas las precauciones por aerosoles."), nl
    ; write(Personal), write(" NO cumple todas las precauciones por aerosoles."), nl,
    mostrar_faltantes(Personal, aerosoles)
    ).
% Mostrar faltantes
mostrar_faltantes(Personal, Categoria) :-

```

```

obtener_precauciones(Categoria, Lista),
findall(Prec, (member(Prec, Lista), \+ aplica(Personal, Prec)), Faltantes),
write("Faltan las siguientes medidas: "), write(Faltantes), nl.
% Obtener lista según categoría
obtener_precauciones(estandar, Lista) :-
findall(Prec, precaucion_estandar(Prec), Lista).
obtener_precauciones(gotas, Lista) :-
findall(Prec, precaucion_gotas(Prec), Lista).
obtener_precauciones(aerosoles, Lista) :-
findall(Prec, precaucion_aerosoles(Prec), Lista).
%% Aislamiento inferido %%
aislamiento_inferido(Persona, Tipo) :-
casoConfirmado(Persona),
nivel_atencion(Persona, tercer_nivel),
Tipo = hospitalario_individual,
format("~w requiere aislamiento hospitalario individual por ser caso confirmado en te
aislamiento_inferido(Persona, Tipo) :-
casoSospechoso(Persona),
nivel_atencion(Persona, primer_nivel),
Tipo = domicilio,
format("~w debe permanecer en aislamiento domiciliario por ser caso sospechoso sin ne
aislamiento_inferido(Persona, sin_dato) :-
format("No se pudo inferir el tipo de aislamiento para ~w. Verifique el estado del ca
%% Notificar el caso %%
notificar_UIES(Persona) :-
casoSospechoso(Persona),
format("Notificando de inmediato a UIES y jurisdicción sanitaria sobre ~w...~n", [Per
write("Correo: ncov@dgepi.salud.gob.mx"), nl,
write("Teléfonos: 5337-1845 / 800-00-44-800"), nl,
write("Formato: SUIVE-1 con Epiclave No. 191"), nl, !.
notificar_UIES(Persona) :-
write("No se puede notificar porque "), write(Persona), write(" no es un caso sospech
%% realizar seguimiento %%
realizar_seguimiento(Persona) :-
casoConfirmado(Persona),
contacto_con(Otro, Persona),
format("~w tuvo contacto con el caso confirmado ~w.~n", [Otro, Persona]),
write("Se debe iniciar monitoreo diario y estudio epidemiológico a la persona en cont
realizar_seguimiento(Persona) :-
\+ casoConfirmado(Persona),

```

```

format("~w no es un caso confirmado, no se inicia seguimiento de contactos.~n", [Persona],
realizar_seguimiento(Persona) :-
casoConfirmado(Persona),
\+ contacto_con(_, Persona),
format("No se registraron contactos directos con ~w para seguimiento.~n", [Persona]),
%% Muestra tomada %%
muestra_valida(Persona) :-
toma_muestra(_, Persona, exudado_nasofaringeo),
toma_muestra(_, Persona, exudado_faringeo),
format("La muestra de ~w incluye exudado nasofaríngeo y faríngeo: muestra válida para ~w", [Persona]),
muestra_valida(Persona) :-
format("La muestra de ~w NO cumple con los requisitos estándar de diagnóstico (faltan ~w)", [Persona]),
% Clasificación exhaustiva
persona_clasificada(X) :-
(casoSospechoso(X); casoConfirmado(X)),
format("~w está clasificado en el sistema (sospechoso o confirmado).~n", [X]), !.
persona_clasificada(X) :-
format("Alerta: ~w no está clasificado como sospechoso ni confirmado.~n", [X]),
fail.

```

Recordando que esta base del conocimiento y pseudo sistema experto va más orientado a la detección de personas que pueden ser potencialmente casos sospechosos o confirmados, se diseñó el sistema partiendo del hecho **persona/1** para identificar a los sujetos como personas, lo cual es un primer requisito para comenzar con la identificación del caso en cuestión. Se define la relación **presenta\_sintomas/2** que permite expresar que una persona (juan por ejemplo) tiene un sintoma como lo puede ser la fiebre. Otra relación muy necesaria y vital para el contexto es si la persona viajó y a donde para identificar si puede ser un posible caso sospechoso por haber estado en un país de riesgo, esto se expresa mediante la relación **viajo\_a/2** para decir quien y a donde se hizo el viaje. Más adelante se listan los países de riesgo mediante el hecho **pais\_riesgo/1**. Se toma en cuenta la existencia de un antecedente de contacto entre 2 personas mediante la relación **contacto\_directo/2** para especificar que una persona estuvo en contacto con otra potencialmente infectada.

Se incluyeron también mediante hechos las precauciones de higiene que se deben respetar, personal médico y sus distintos tipos de jerarquías relevantes para este caso, el nivel de atención al que acudieron las personas y los tipos de muestras de laboratorio obtenidas.



## 2. Ejemplos de cada caso de las reglas terminológicas vistas en clases.

Ejemplifique cada caso de regla terminológica vista en clase (rc-slides-05, 77) con las definiciones de su representación. (20/100).

Se ejemplifica un caso por cada tipo de regla terminológica de las vistas en clase:

1. Predicados disjuntos.

```
% Ejemplo de caso disjunto
esCasoSospechoso(Persona) :-
    casoConfirmado(Persona),
    format("Error: ~w ya es un caso confirmado, no puede registrarse como sospechoso\n", Persona), !, fail.

esCasoSospechoso(Persona) :-
    casoSospechoso(Persona),
    write(Persona), write(" ya está registrado como caso sospechoso."), nl, !.
```

Pese a que no se ejemplifica explícitamente y de forma más sencilla como decir que un caso sospechoso es un caso no confirmado por querer evitar el uso de la negación en el sistema, muestra implícitamente que un caso sospechoso NO puede ser un caso confirmado, porque para ser confirmado primero debe ser un caso sospechoso, entonces la primer clausula funge como un checador de que no suceda esto.

2. Subtipo.

```
%% Tipo %%
personal_salud(X) :- enfermera(X).
personal_salud(X) :- epidemiologo(X).
%% Subtipo %%
enfermera(enfermera1).
epidemiologo(epidemiol1).
```

Aquí se muestra de forma más explícita como la clase personal de salud subsume a las subclases de enfermera y epidemiologo.

### 3. Exhaustivo.

```
% Clasificación exhaustiva
persona_clasificada(X) :-
(casoSospechoso(X); casoConfirmado(X)),
format("~w está clasificado en el sistema (sospechoso o confirmado).~n", [X]), !
```

Aquí se muestra como el ejemplo de caso exhaustivo al tomar en cuenta si una persona es clasificada o no, y para hecho, la persona debe ser asignada como caso confirmado O sospechoso.

### 4. Simétrico

```
contacto_directo(juan, maria).
contacto_con(A, B) :-
contacto_directo(A, B);
contacto_directo(B, A).
```

La definición **contacto\_con/2** muestra la relación del contacto directo como una definición simétrica.

### 5. Inverso

```
% Verificar precaucion aplicada
precaucion_aplicada_a(Precaucion, Personal) :-
aplica(Personal, Precaucion).
```

Esta definición muestra la dualidad del inverso, representando si la precaución ya fue aplicada, por quien y biceversa.

6. Restricción.

```
muestra_valida(Persona) :-  
toma_muestra(_, Persona, exudado_nasofaringeo),  
toma_muestra(_, Persona, exudado_faringeo),  
format("La muestra de ~w incluye exudado nasofaríngeo y faríngeo: muestra válida"  
  
muestra_valida(Persona) :-  
format("La muestra de ~w NO cumple con los requisitos estándar de diagnóstico (f
```

Aquí se muestra como **muestra\_valida/1** tiene condiciones necesarias para que la muestra la pueda considerar una muestra valida, fungiendo como restricción.

7. Definición.

```
esCasoConfirmado(Persona) :-  
esCasoSospechoso(Persona),  
resultado_laboratorio(Persona, positivo),  
format("~w es un caso confirmado porque su prueba de laboratorio resultó positiva"  
asserta(casoConfirmado(Persona)), !.
```

Se muestra la definición de un caso confirmado y los criterios que debe cumplir para que sea verdadero.

### 3. Ejemplos de las preguntas que puede resolver la representación

**Ejemplifique y justifique el tipo de preguntas que su representación y Prolog pueden resolver. (30/100)**

El programa, como ya se menciono anteriormente, busca enfocarse en el diagnostico de pacientes, por lo que haciendo uso de un acomodo bonito de las consultas a prolog, se le pueden hacer principalmente, preguntas relacionadas a la identificación y diagnostico de personas que pueden ser casos confirmados o no confirmados.

Dentro de las principales preguntas que se le pueden hacer a este pseudo sistema experto, son: Si pedro es un caso sospechoso, si es un caso confirmado, si debería ser aislado, si es necesario notificar al UIES, si las muestras obtenidas son validas, entre otras particularidades.

Otras cosas que se pueden preguntar son cosas como cual es la lista de países riesgosos o cuales son las precauciones que se deben tomar.

```
% 1: Evaluar si Juan es caso sospechoso
```

```
esCasoSospechoso(juan).
```

```
juan cumple los criterios de caso sospechoso por viaje reciente a país de riesgo: chi
```

```
% 2: Evaluar si Juan es caso confirmado
```

```
?- esCasoConfirmado(juan)
```

```
juan cumple los criterios de caso sospechoso por viaje reciente a país de riesgo: chi
```

```
juan es un caso confirmado porque su prueba de laboratorio resultó positiva.
```

```
% 3.- Inferir el tipo de aislamiento para Juan
```

```
?- aislamiento_inferido(juan, Tipo).
```

```
juan debe permanecer en aislamiento domiciliario por ser caso sospechoso sin necesidad
```

```
Tipo = domicilio.
```

```
% 4: Verificar si se debe notificar a UIES
```

```
?- notificar_UIES(juan).
```

```
Notificando de inmediato a UIES y jurisdicción sanitaria sobre juan...
```

```
Correo: ncov@dgepi.salud.gob.mx
```

```
Teléfonos: 5337-1845 / 800-00-44-800
```

```
Formato: SUIVE-1 con Epiclave No. 191
```

```
true.
```

```

% 5: Verificar si debe hacerse seguimiento a contactos
?- realizar_seguimiento(maria).
maria no es un caso confirmado, no se inicia seguimiento de contactos.
false.

% 6: Validar si la muestra tomada a Juan es válida
?- muestra_valida(juan).
La muestra de juan incluye exudado nasofaríngeo y faríngeo: muestra válida para diagn
true.

% 7: ¿Está clasificado como sospechoso o confirmado?
?- persona_clasificada(juan).
juan está clasificado en el sistema (sospechoso o confirmado).
true.

% 8: ¿Se puede clasificar a alguien no registrado aún?
?- persona_clasificada(pedro).
Alerta: pedro no está clasificado como sospechoso ni confirmado.
false.

% 9: Cumple con las precauciones estandar
cumple_precauciones_estandar(enfermera1).
enfermera1 NO cumple todas las precauciones estándar.
Faltan las siguientes medidas: [uso_mascarilla,uso_contenedores]
true.

```

Estas consultas ejemplifican lo que puede hacer en su mayoría el sistema, resulta curioso porque para realizar algunas, es necesario dotarle de consultas previas que son capaces de añadir información a la base del conocimiento.

## **4. Referencias**

### **Referencias**

Guerra-Hernández, A. (2025). Deducción natural. Notas de clase, Capítulo 5, Curso de Representación del Conocimiento, Universidad Veracruzana.