# Triage Respiratorio en Prolog — Guía Profesional con Consultas

**Autor:** M.C.I. Francisco Primero

## Resumen ejecutivo

Este documento describe un sistema determinista de triaje respiratorio escrito en Prolog. Incluye el código completo, la explicación de cada componente, el flujo de decisión, consultas profesionales con salidas esperadas, técnicas de depuración (trace), y extensiones sugeridas.

## Código completo (triage\_respiratorio.pl)

% ========================  
% TRIAJE RESPIRATORIO  
% (reglas deterministas)  
% ========================  
  
% ---- Hechos de pacientes (ejemplo) ----  
sintoma(juan, fiebre).  
sintoma(juan, tos).  
sintoma(juan, dolor\_muscular).  
  
sintoma(maria, estornudos).  
sintoma(maria, congestion).  
sintoma(maria, picazon\_ojos).  
  
sintoma(pedro, disnea\_severa). % alerta  
sintoma(pedro, tos).  
  
sintoma(lucia, fiebre).  
sintoma(lucia, tos\_seca).  
sintoma(lucia, dolor\_garganta).  
  
edad(juan, 32).  
edad(maria, 27).  
edad(pedro, 71).  
edad(lucia, 45).  
  
comorbilidad(pedro, epoc).  
  
% ---- 1) ALERTAS (cortan la decisión) ----  
alerta\_sintoma(disnea\_severa).  
alerta\_sintoma(dolor\_toracico).  
alerta\_sintoma(sat\_oxigeno\_baja).  
  
tiene(P, S) :- sintoma(P, S).  
  
detectar\_alerta(P, S) :-  
 alerta\_sintoma(S),  
 tiene(P, S), !. % ¡corte!: si hay alerta, se termina el flujo  
  
% ---- 2) VULNERABILIDAD ----  
vulnerable(P) :-  
 edad(P, E), E >= 65, !.  
vulnerable(P) :-  
 comorbilidad(P, \_).  
  
% ---- 3) CLASIFICACIÓN (conteo simple) ----  
requisito(gripe, fiebre).  
requisito(gripe, tos).  
requisito(gripe, dolor\_muscular).  
  
requisito(resfriado, congestion).  
requisito(resfriado, estornudos).  
requisito(resfriado, tos).  
  
requisito(covid, fiebre).  
requisito(covid, tos\_seca).  
requisito(covid, dolor\_garganta).  
  
requisito(alergia, congestion).  
requisito(alergia, estornudos).  
requisito(alergia, picazon\_ojos).  
  
contraindica(alergia, fiebre).  
  
puntaje(P, Dx, N) :-  
 findall(S, (requisito(Dx, S), tiene(P, S)), A), length(A, A1),  
 findall(S, (contraindica(Dx, S), tiene(P, S)), C), length(C, C1),  
 N is A1 - C1.  
  
diagnostico(P, Dx, N) :- puntaje(P, Dx, N), N > 0.  
  
max\_puntaje(P, Max) :-  
 findall(N, diagnostico(P, \_, N), Ns),  
 Ns \= [], max\_list(Ns, Max).  
  
mejores\_dx(P, Dxs) :-  
 max\_puntaje(P, M),  
 findall(Dx, diagnostico(P, Dx, M), Dxs).  
  
% ---- 4) PLAN ----  
plan\_para(urgente(\_), derivar\_urgencias).  
plan\_para(probable(alergia), antihistaminico).  
plan\_para(probable(resfriado), reposo\_hidratacion).  
plan\_para(probable(gripe), aines\_reposo).  
plan\_para(probable(covid), test\_covid\_aislamiento).  
  
% Afecta plan si es vulnerable  
ajustar\_por\_vulnerable(Plan, true, control\_48h(Plan)).  
ajustar\_por\_vulnerable(Plan, false, Plan).  
  
% ---- Orquestador: secuencia de ideas ----  
evaluar\_paciente(P, reporte{  
 riesgo: Riesgo,  
 diagnosticos: DxList,  
 plan: PlanAjustado,  
 notas: Notas  
}) :-  
 ( detectar\_alerta(P, S) ->  
 Riesgo = urgente,  
 DxList = [],  
 plan\_para(urgente(S), Plan),  
 ajustar\_por\_vulnerable(Plan, false, PlanAjustado),  
 Notas = [alerta(S), "derivar de inmediato"]  
 ; Riesgo = no\_urgente,  
 (vulnerable(P) -> Vul = true ; Vul = false),  
 ( mejores\_dx(P, Dxs) -> DxList = Dxs ; DxList = [] ),  
 decidir\_plan(DxList, Vul, PlanAjustado, Notas)  
 ).  
  
decidir\_plan([Dx], Vul, PlanAjustado, [diagnostico\_unico(Dx)]) :-  
 plan\_para(probable(Dx), Plan),  
 ajustar\_por\_vulnerable(Plan, Vul, PlanAjustado), !.  
  
decidir\_plan([Dx1, Dx2 | \_], Vul, PlanAjustado, [diagnostico\_ambiguo([Dx1, Dx2])]) :-  
 % si hay empate, elige uno neutral: test si incluye covid, si no, manejo sintomático  
 ( member(covid, [Dx1, Dx2])  
 -> plan\_para(probable(covid), Base)  
 ; Base = reposo\_hidratacion),  
 ajustar\_por\_vulnerable(Base, Vul, PlanAjustado), !.  
  
decidir\_plan([], \_, sin\_datos, ["no hay suficientes sintomas"]).

## Cómo cargar y ejecutar

1) Guarde el archivo como triage\_respiratorio.pl

2) Cargue con ruta absoluta (Windows):

?- consult('C:/Users/Frank Prime/Documents/UICUI/5° I.A/triage\_respiratorio.pl').

"C:\Users\rodri\OneDrive\Escritorio\fallas\_carros.pl"

3) Recargue tras editar:

?- make.

4) Salir:

?- halt.

## Consultas profesionales — escenarios completos

A) Paciente JUAN — fiebre, tos, dolor\_muscular; 32 años, sin comorbilidad.

Consulta:

?- evaluar\_paciente(juan, R).

Salida esperada (estructural):

R = reporte{riesgo:no\_urgente, diagnosticos:[gripe], plan:aines\_reposo, notas:[diagnostico\_unico(gripe)]}.

Verificación del razonamiento:

?- findall(Dx-N, diagnostico(juan, Dx, N), L). % L = [gripe-3, resfriado-1, covid-1]

?- mejores\_dx(juan, Dxs). % Dxs = [gripe]

B) Paciente MARIA — estornudos, congestion, picazon\_ojos; 27 años, sin comorbilidad.

Consulta:

?- evaluar\_paciente(maria, R).

Salida esperada:

R = reporte{riesgo:no\_urgente, diagnosticos:[alergia], plan:antihistaminico, notas:[diagnostico\_unico(alergia)]}.

Verificación:

?- findall(Dx-N, diagnostico(maria, Dx, N), L). % L = [alergia-3, resfriado-2]

?- mejores\_dx(maria, Dxs). % Dxs = [alergia]

C) Paciente PEDRO — disnea\_severa (alerta), tos; 71 años, EPOC.

Consulta:

?- evaluar\_paciente(pedro, R).

Salida esperada:

R = reporte{riesgo:urgente, diagnosticos:[], plan:derivar\_urgencias, notas:[alerta(disnea\_severa), "derivar de inmediato"]}.

Verificación de alerta (corte):

?- detectar\_alerta(pedro, S). % S = disnea\_severa

D) Paciente LUCIA — fiebre, tos\_seca, dolor\_garganta; 45 años, sin comorbilidad.

Consulta:

?- evaluar\_paciente(lucia, R).

Salida esperada:

R = reporte{riesgo:no\_urgente, diagnosticos:[covid], plan:test\_covid\_aislamiento, notas:[diagnostico\_unico(covid)]}.

Verificación:

?- findall(Dx-N, diagnostico(lucia, Dx, N), L). % L = [covid-3, gripe-1]

?- mejores\_dx(lucia, Dxs). % Dxs = [covid]

## Consultas de apoyo (diagnóstico, empates, plan)

Puntajes detallados de un paciente:

?- findall(Dx-N, diagnostico(juan, Dx, N), L).

Máximo puntaje y mejores diagnósticos:

?- max\_puntaje(juan, M), mejores\_dx(juan, Dxs).

Política en caso de empate (ejemplo genérico):

?- decidir\_plan([covid, resfriado], false, Plan, Notas).

## Trazado (trace) para seguir la secuencia

Activar y desactivar trace alrededor de una evaluación:

?- trace, evaluar\_paciente(juan, R), notrace.

Verás el orden: detectar\_alerta → vulnerable → puntaje/diagnostico → max\_puntaje → mejores\_dx → decidir\_plan.

## Depuración rápida y errores comunes

* Cada consulta debe terminar con punto: `.`
* Si editas el archivo, recarga con `?- make.`
* Si Prolog dice que no encuentra el archivo, usa ruta absoluta con `consult('C:/ruta/triage\_respiratorio.pl').`
* Si ves `true.` y no ves valores, usa `findall/3` para materializar listas y resultados intermedios.

## Extensiones sugeridas (no IA, siguen siendo reglas)

1) Pesos por síntoma: `peso(Dx, Sintoma, W)` y reemplazar `length/2` por suma de pesos.

2) Síntomas negativos relevantes (ausencia) como contradicción explícita.

3) Reglas duras de exclusión (ej., saturación baja → urgente sin pasar por clasificador).

4) Explicabilidad: `explicacion(P, Dx, A\_Favor, En\_Contra)` devolviendo listas de evidencias.