Pontificia Universidad
JAVERIANA
Bogotá

Introducción a IA

Taller 6

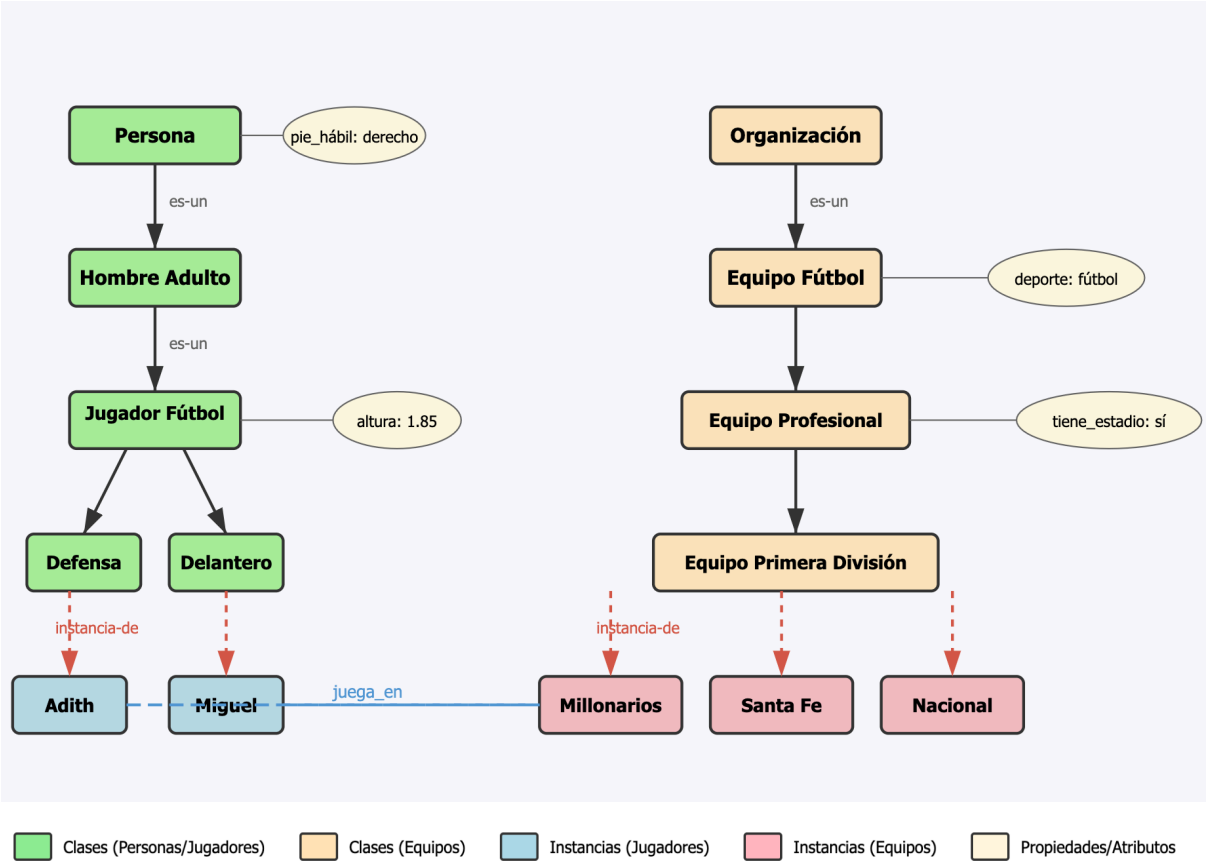
Grupo 5

Santos Alejandro Arellano Olarte // **Santos-arellano**
Alejandra Abaunza Suárez
Daniel Santiago Avila Medina
Jeison Camilo Alfonso Moreno

11/ Septiembre /2025

Descripción del Modelo

Esta red semántica representa el conocimiento sobre jugadores de fútbol y equipos, implementando un sistema de herencia de propiedades mediante relaciones **es-un** (jerarquía de clases) e **instancia-de** (pertenencia a clases).



Ampliación Implementada

Nueva Jerarquía de Equipos: Se ha añadido una rama completa para representar equipos de fútbol con las siguientes características:

- **Organización** → Clase raíz para entidades organizativas
- **Equipo Fútbol** → Especialización deportiva
- **Equipo Profesional** → Equipos con estructura profesional
- **Equipo Primera División** → Máxima categoría
- **Instancias:** Millonarios, Santa Fe, Nacional

Los jugadores (Miguel y Adith) están vinculados al equipo Millonarios mediante la relación juega_en.

Funcionalidades Implementadas

- Herencia múltiple de propiedades a través de la jerarquía es-un
- Resolución de conflictos: las propiedades más específicas sobrescriben las heredadas
- Consultas bidireccionales: desde instancias hacia clases y viceversa
- Relaciones entre diferentes jerarquías (jugador-equipo)
- Sistema de inferencia para deducir propiedades no explícitas

Codigo

```
% ================================================
% TALLER 6 - REDES SEMÁNTICAS EN PROLOG
% ================================================

% ================================================
% PARTE 1: DEFINICIÓN DE LA RED SEMÁNTICA BASE
% ================================================

% Relaciones de herencia (es_un)
es_un(hombre_adulto, persona).
es_un(jugador_futbol, hombre_adulto).
es_un(defensa, jugador_futbol).
es_un(delantero, jugador_futbol).

% Relaciones de instancia (instancia_de)
instancia_de(miguel, delantero).
```

```
instancia_de(adith, defensa).

% ================================================
% PARTE 2: PROPIEDADES Y ATRIBUTOS
% ================================================

% Propiedades directas de las clases
propiedad(persona, pie_habil, derecho).
propiedad(hombre_adulto, altura, 1.80).
propiedad(jugador_futbol, altura, 1.85).
propiedad(jugador_futbol, numero_goles, 3).
propiedad(jugador_futbol, patea, balon).
propiedad(defensa, numero_goles, 1).
propiedad(delantero, numero_goles, 5).

% Propiedades específicas de instancias
propiedad(miguel, equipo, millonarios).
propiedad(adith, equipo, millonarios).

% ================================================
% PARTE 3: MECANISMO DE HERENCIA
% ================================================

% Regla para verificar si X es subclase de Y (transitivo)
subclase_de(X, Y) :- es_un(X, Y).
subclase_de(X, Y) :-
   es_un(X, Z),
   subclase_de(Z, Y).

% Regla para verificar si X pertenece a la clase Y
pertenece_a(X, Y) :- instancia_de(X, Y).
pertenece_a(X, Y) :-
   instancia_de(X, Z),
   subclase_de(Z, Y).
```

```
% ================================================
% PARTE 4: HERENCIA DE PROPIEDADES
% ================================================

% Obtener valor de una propiedad con herencia
obtener_propiedad(Objeto, Atributo, Valor) :-
   % Primero busca si el objeto tiene la propiedad directamente
   propiedad(Objeto, Atributo, Valor), !.

obtener_propiedad(Objeto, Atributo, Valor) :-
   % Si es una instancia, busca en su clase
   instancia_de(Objeto, Clase),
   obtener_propiedad_clase(Clase, Atributo, Valor), !.

obtener_propiedad(Clase, Atributo, Valor) :-
   % Si es una clase, busca en sus superclases
   es_un(Clase, Superclase),
   obtener_propiedad_clase(Superclase, Atributo, Valor).

% Buscar propiedad en una clase o sus superclases
obtener_propiedad_clase(Clase, Atributo, Valor) :-
   propiedad(Clase, Atributo, Valor), !.

obtener_propiedad_clase(Clase, Atributo, Valor) :-
   es_un(Clase, Superclase),
   obtener_propiedad_clase(Superclase, Atributo, Valor).

% ================================================
% PARTE 5: CONSULTAS ÚTILES
% ================================================

% Listar todas las propiedades de un objeto
listar_propiedades(Objeto) :-
```

```
   write('Propiedades de '), write(Objeto), write(':'), nl,
   forall(obtener_propiedad(Objeto, Atributo, Valor),
          (write('  '), write(Atributo), write(' = '), write(Valor), nl)).

% Verificar si un objeto puede realizar una acción
puede(Objeto, Accion, Sobre) :-
   obtener_propiedad(Objeto, Accion, Sobre).

% Obtener todos los objetos de una clase
objetos_de_clase(Clase, Objeto) :-
   pertenece_a(Objeto, Clase).

% ================================================
% PARTE 6: AMPLIACIÓN CON JERARQUÍA DE EQUIPOS
% ================================================

% Nueva jerarquía para equipos de fútbol
es_un(equipo_futbol, organizacion).
es_un(equipo_profesional, equipo_futbol).
es_un(equipo_primera_division, equipo_profesional).

% Instancias de equipos
instancia_de(millonarios, equipo_primera_division).
instancia_de(santa_fe, equipo_primera_division).
instancia_de(nacional, equipo_primera_division).

% Propiedades de la jerarquía de equipos
propiedad(organizacion, tipo, deportiva).
propiedad(equipo_futbol, deporte, futbol).
propiedad(equipo_futbol, numero_jugadores, 11).
propiedad(equipo_profesional, tiene_estadio, si).
propiedad(equipo_primera_division, division, primera).

% Propiedades específicas de equipos
```

```
propiedad(millonarios, ciudad, bogota).
propiedad(millonarios, fundacion, 1946).
propiedad(millonarios, colores, 'azul y blanco').
propiedad(santa_fe, ciudad, bogota).
propiedad(santa_fe, fundacion, 1941).
propiedad(santa_fe, colores, 'rojo y blanco').
propiedad(nacional, ciudad, medellin).
propiedad(nacional, fundacion, 1947).
propiedad(nacional, colores, 'verde y blanco').

% Relación jugador-equipo
juega_en(Jugador, Equipo) :-
   obtener_propiedad(Jugador, equipo, Equipo).

% Jugadores del mismo equipo
companeros(Jugador1, Jugador2) :-
   juega_en(Jugador1, Equipo),
   juega_en(Jugador2, Equipo),
   Jugador1 \= Jugador2.

% ================================================
% PARTE 7: CONSULTAS DE DEMOSTRACIÓN
% ================================================

% Demostrar el funcionamiento del sistema
demo :-
   nl, write('=== DEMOSTRACIÓN DE REDES SEMÁNTICAS ==='), nl, nl,

   write('1. Herencia de propiedades:'), nl,
   write('   Altura de Adith: '),
   obtener_propiedad(adith, altura, AlturaAdith),
   write(AlturaAdith), nl,

   write('   Pie hábil de Miguel: '),
```

```
    obtener_propiedad(miguel, pie_habil, PieMiguel),
    write(PieMiguel), nl, nl,

    write('2. Propiedades específicas vs heredadas:'), nl,
    write('   Goles de un defensa genérico: '),
    obtener_propiedad(defensa, numero_goles, GolesDefensa),
    write(GolesDefensa), nl,
    write('   Goles de un delantero genérico: '),
    obtener_propiedad(delantero, numero_goles, GolesDelantero),
    write(GolesDelantero), nl, nl,

    write('3. Verificación de acciones:'), nl,
    write('   ¿Puede Miguel patear un balón? '),
    (puede(miguel, patea, balon) -> write('Sí') ; write('No')), nl, nl,

    write('4. Compañeros de equipo:'), nl,
    write('   ¿Son Miguel y Adith compañeros? '),
    (companeros(miguel, adith) -> write('Sí') ; write('No')), nl, nl,

    write('5. Propiedades del equipo Millonarios:'), nl,
    listar_propiedades(millonarios), nl,

    write('6. Todas las propiedades de Miguel:'), nl,
    listar_propiedades(miguel).

% ================================================
% CONSULTAS DE EJEMPLO
% ================================================

% Para ejecutar las consultas:
% ?- demo.
% ?- obtener_propiedad(adith, altura, X).
% ?- obtener_propiedad(miguel, pie_habil, X).
% ?- puede(miguel, patea, balon).
```

```
% ?- companeros(miguel, adith).
% ?- listar_propiedades(miguel).
% ?- listar_propiedades(millonarios).
% ?- objetos_de_clase(jugador_futbol, X).
% ?- objetos_de_clase(equipo_primera_division, X).
```