UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA
ESCUELA DE CIENCIAS Y SISTEMAS
LABORATORIO DE MODELACION Y SIMULACION 2.

PRACTICA 2

MANUAL TECNICO

Nombres	Carné
Mario Jose Rodriguez Vasquez	201908338
Erick Ivan Mayorga Rodriguez	201901758

Guatemala, 15 de Junio de 2024.

Introduccion

Este documento describe la implementación de una red semántica utilizando Prolog para representar una variedad de videojuegos, organizados en géneros, y sus respectivas propiedades. También se presenta un diagrama que simplifica las relaciones entre los diferentes elementos de la red semántica.

Objetivo General

- Implementar una red semántica para videojuegos utilizando Prolog.

Objetivos Especificos

- Representar diferentes generos de videojuegos.
- Relacionar videojuegos especificos con sus generos.
- Asignar propiedaes a cada videojuego.
- Simplificar las relaciones en un diagrama visual claro.

Estructura de Codigo

El código Prolog está dividido en tres secciones principales: hechos, reglas y consultas de ejemplo.

Hechos

Los hechos definen las instancias y subclases, así como las propiedades de cada videojuego.

```
%%%%%%HECHOS
%instancia_de(Objeto, Clase)
instancia_de(super_mario, plataforma_juego).
instancia_de(mario_kart, carreras).
instancia_de(zelda, aventura).
instancia_de(pokemon, rpg).
instancia_de(call_of_duty, fps).
instancia_de(fortnite, battle_royale).
instancia_de(overwatch, fps).
instancia_de(minecraft, sandbox).
instancia_de(the_sims, simulacion).
instancia_de(civilization, estrategia).

instancia_de(halo, fps).
instancia_de(forza_horizon, carreras).

%subclase(Subclase, Clase)
subclase_de(plataforma_juego, videojuegos).
subclase_de(carreras, videojuegos).
subclase_de(aventura, videojuegos).
subclase_de(rpg, videojuegos).
subclase_de(fps, videojuegos).
subclase_de(battle_royale, videojuegos).
subclase_de(sandbox, videojuegos).
subclase_de(simulacion, videojuegos).
subclase_de(estrategia, videojuegos).
```

```
%tiene_propiedad(Instancia, Propiedad, Valor)
tiene_propiedad(super_mario, plataforma, nintendo).
tiene_propiedad(super_mario, lanzado, 1985).
tiene_propiedad(mario_kart, plataforma, nintendo).
tiene_propiedad(mario_kart, lanzado, 1992).
tiene_propiedad(zelda, plataforma, nintendo).
tiene_propiedad(zelda, lanzado, 1986).
tiene_propiedad(pokemon, plataforma, nintendo).
tiene_propiedad(pokemon, lanzado, 1996).
      You, 2 minutes ago • Uncommitted changes
tiene_propiedad(call_of_duty, plataforma, multiplataforma).
tiene_propiedad(call_of_duty, lanzado, 2003).
tiene_propiedad(fortnite, plataforma, multiplataforma).
tiene_propiedad(fortnite, lanzado, 2017).
tiene_propiedad(overwatch, plataforma, multiplataforma).
tiene_propiedad(overwatch, lanzado, 2016).
tiene_propiedad(minecraft, plataforma, multiplataforma).
tiene_propiedad(minecraft, lanzado, 2011).
tiene_propiedad(the_sims, plataforma, multiplataforma).
tiene_propiedad(the_sims, lanzado, 2000).
tiene_propiedad(civilization, plataforma, multiplataforma).
tiene_propiedad(civilization, lanzado, 1991).

tiene_propiedad(halo, plataforma, xbox).
tiene_propiedad(halo, lanzado, 2001).
tiene_propiedad(forza_horizon, plataforma, xbox).
tiene_propiedad(forza_horizon, lanzado, 2012).
```

Reglas

Las reglas definen la lógica para determinar si un objeto es una instancia de una clase, para conocer las propiedades de una instancia y para verificar si una clase es subclase de otra.

```
%%%%%%REGLAS
% Regla para saber si un objeto es una instancia de una clase en concreto
es_instancia_de(Instancia, Clase) :-
    instancia_de(Instancia, Clase).
es_instancia_de(Instancia, Clase) :-
    instancia_de(Instancia, SubClase),
    subc(SubClase, Clase).

% Regla para saber si una clase es subclase de otra
subc(SubClase, Clase) :-
    subclase_de(SubClase, Clase).
subc(SubClase, Clase) :-
    subclase_de(SubClase, ClaseIntermedia),
    subc(ClaseIntermedia, Clase).

% Regla para conocer todas las propiedades que posee una instancia
propiedades_de(Instancia, Propiedad, Valor) :-
    tiene_propiedad(Instancia, Propiedad, Valor).
```

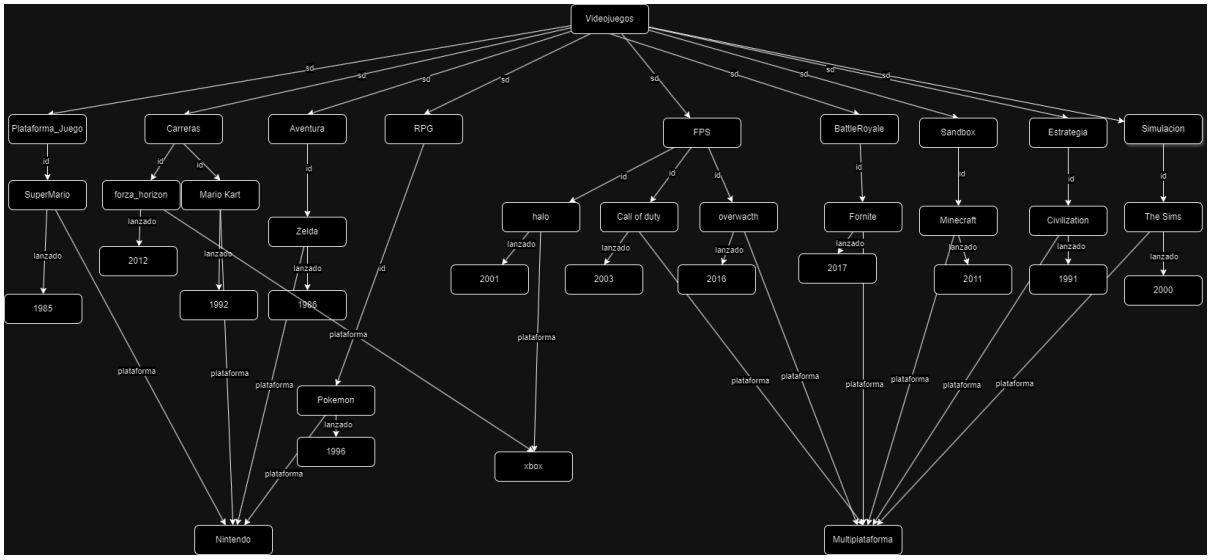
Consultas de Ejemplo

Las consultas de ejemplo muestran cómo usar las reglas para obtener información de la red semántica.

```
% Consultas de ejemplo
% ¿Super Mario es una instancia de Videojuegos?
% es_instancia_de(videojuegos, super_mario).

% ¿En qué plataforma fue lanzado Super Mario?
% tiene_propiedad(super_mario, plataforma, X).

%%% ¿Plataforma_juego es una subclase de Videojuegos?
% subc(plataforma_juego, videojuegos).

% clases de las que es instancia super Mario
%es_instancia_de(Y,super_mario).
```

Diagrama de Red Semántica

El diagrama simplificado de la red semántica muestra cómo se relacionan los videojuegos con sus géneros y plataformas de una manera clara y concisa.



Explicacion del Diagrama

- **Nodo Central**: El nodo "Videojuegos" es el punto de partida.
- **Géneros**: Cada género (Plataforma, Carreras, Aventura, etc.) se conecta al nodo central.
- **Videojuegos Específicos**: Cada género tiene videojuegos específicos conectados.
- **Plataformas**: Los videojuegos que comparten la misma plataforma apuntan a un solo nodo para esa plataforma (por ejemplo, Nintendo o Multiplataforma).
- **Años de Lanzamiento**: Cada videojuego tiene su año de lanzamiento especificado.