Trabajo Práctico 2

Agustín Díaz Aldana Zarate

1 - Descripción del dominio

Videojuegos

Un videojuego puede ser de un solo jugador o multijugador. Un multijugador se puede jugar en modo cooperativo y/o en versus. Un videojuego además posee una puntuación, una fecha de creación, una dimensión (2d, 2.5d o 3d) y una Plataforma (PC, Mobile o Console). Un videojuego es desarrollado por un estudio formado por programadores y diseñadores. Estos estudios pueden ser independientes (recién están comenzando) o pueden ser ya grandes corporaciones.

Un videojuego puede tener o no logros (que los va ganando el jugador a medida que va avanzando en el juego).

Al final de cada año, un juego puede ganar diferentes premios.

Para mejorar la experiencia, cada videojuego posee una banda sonora acorde al tipo del mismo. Dicha banda sonora está conformada por diferentes músicos.

Un género de videojuego puede ser de varios tipos como Puzzle, Plataforma, RPG, Runner, Souls-like y muchos más. En particular los juegos del tipo género Rogue pueden ser RogueLike o RogueLite y los de Simulación pueden ser de Auto, Avión, Cocina o Granja.

Para visualizar la ontología asociada a la descripción anterior, ver el archivo **Diaz-Zarate-Ontologia.pdf**

2- Descripción del dominio en Protegé

Ver archivo **Diaz-Zarate.owl**
Nota: no se agregó la restricción equivalente implementada en Independiente en Corporación ya que requería la entrada de demasiadas personas, pero la idea es que las corporaciones tengan un mínimo de 50 empleados.

3 - Instancias

Ver archivo **Diaz-Zarate.owl**

4 - Queries

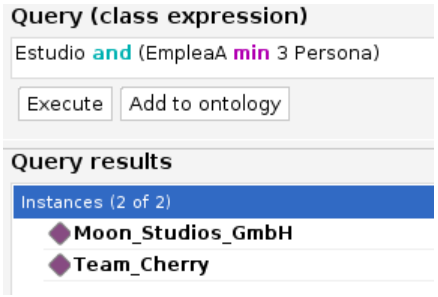
Utilizamos el razonador Fact++

Guillermina

Guillermina está entrando en el mundo de programación. Quiere buscar algún estudio para trabajar pero le interesa que haya un mínimo de 3 personas en él para asegurarse de que va a tener gente que la pueda guiar.
La ayudamos con esta query:

```
Estudio and (EmpleaA min 3 Persona)
```

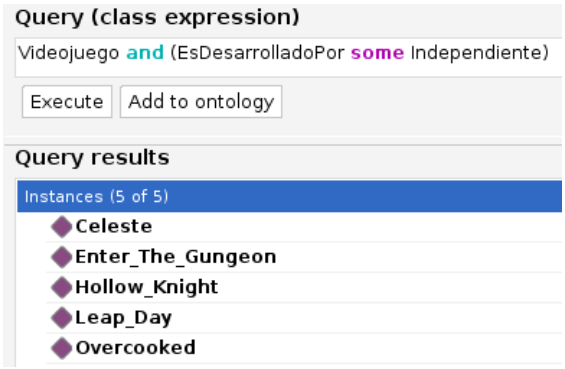
La cual devuelve:



Federico

Mi amigo Federico es un músico indie, y está en la movida anti-corporativa. Se me ocurrió buscarle unos juegos con una query en protege

```
Videojuego and (EsDesarrolladoPor some Independiente)
```



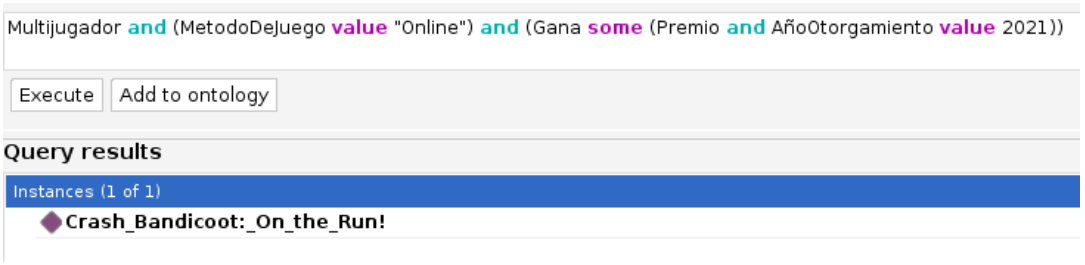
Guadalupe

Mi hermana Guadalupe vio que el gaming está ganando tracción debido a la pandemia. Esto la encaminó en la búsqueda de un juego que le haya ido bien este año y que pueda jugar con sus amigxs

Con los requerimientos de Guadalupe, generamos esta DL Query en protege

```
Multijugador and (MetodoDeJuego value "Online") and (Gana some (Premio
and AñoOtorgamiento value 2021))
```

La cual con los individuos ingresados nos devuelve:



Álvaro

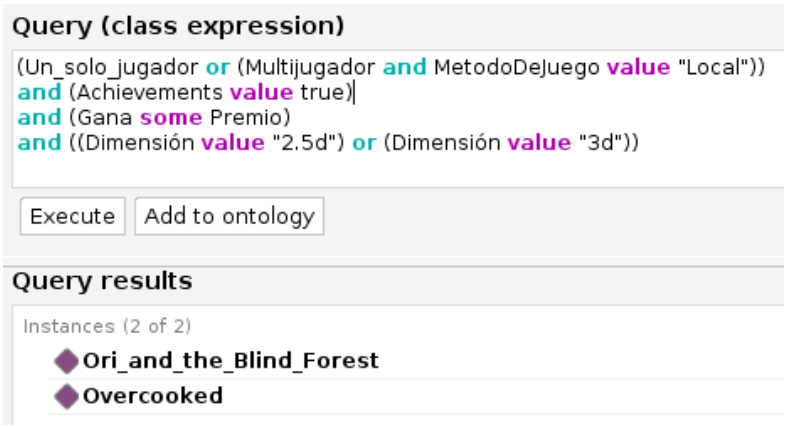
Mi amigo Álvaro está buscando un nuevo juego para jugar.

- Álvaro no está teniendo conexión a internet estable en su zona, por lo cual no puede jugar juegos online.
- Álvaro es lo que llaman “Cazador de logros”, por ende los juegos que adquiere tienen que tener logros excluyentemente.
- Además no le gustan los juegos que se ven muy planos como los de 2 dimensiones.
- Hay que tener en cuenta que Álvaro solo juega lo mejor de lo mejor, y no se guía por los puntajes de la comunidad sino por que los juegos hayan ganado premios.

Con los requerimientos de Álvaro, generamos esta DL Query en protege

```
(Un_solo_jugador or (Multijugador and MetodoDeJuego value "Local"))
and (Achievements value true)
and (Gana some Premio)
and ((Dimensión value "2.5d") or (Dimensión value "3d"))
```

Lo cual nos devuelve:



Conclusiones/Experiencias

Experiencias:

1. Cuando creamos una instancia con restricciones dateTime, lo escribíamos bien pero si no lo formateabamos como dateTime lanzaba una inconsistencia.
2. El punto 1 no pasa con los integer, por ejemplo.
3. Cuando creamos una instancia de multijugador Online, escribimos Online erróneamente y nos lanzó una inconsistencia.
4. Al implementar las restricciones en los estudios, lo hicimos en las subclases en vez de la clase principal Estudio, para evitar inconsistencias/errores que nos aparecían.
5. Protege nos simplificó mucho la parte de entrada de datos gracias al sistema de inferencia del razonador.

Conclusiones:

Notamos que la realidad es muy compleja, lo cual nos fuerza a tomar decisiones sobre el modelado que optamos en base al problema que se quiere resolver. El mundo de los videojuegos es enorme, con lo cual en este trabajo se analizan sólo algunos aspectos del mismo, o al menos las áreas que nos resultan de interés.
La ontología nos promueve a utilizar la relación “is a” gracias a su gran potencial de inferencia y su simplicidad. Sin embargo, mientras más se extienden las relaciones entre clases a modelar, cada vez son más difusas las maneras de crear las subclases para la ontología.
La relación de “is a” permite generar un árbol de relaciones mientras que las relaciones de otros tipos nos permiten generar grafos. Estas distinciones son importantes al momento del modelado ya que por más que la propiedad “is a” sea muy poderosa, rara vez alcanza para modelar algo tan complejo e interrelacionado como la realidad.