

# Estructura RDF basado en el juego DOTA 2

Guevara Tanta, Junior	Hervias Arancibia, Felix	Vera Baldeón Samir	Victor Manuel Saravia Tinco
UNI	UNI	UNI	UNI
Lima, Perú	Lima, Perú	Lima, Perú	Lima, Perú
jguevart@uni.pe	fherviasa@umi.pe	sverab@uni.pe	victor.saravia.t@uni.pe

**Resumen**—Considerando que RDF es una forma general para aislar o separar cualquier tipo de datos en porciones pequeñas, con algunas normas o reglas acerca de la semántica o significado, de esas partes. [1]

En la presente se describirá como usar RDF para describir vocabularios del popular videojuego DOTA2 separando y clasificando conceptos para la creación de un grafo que permita visualizar la descripción de recursos.

## I. INTRODUCCIÓN

RDF se creó en el año 1998 y fue recomendado por W3C en 1999. Es el acrónimo de Resource Description Framework y es un lenguaje para la representación de la información sobre los recursos en la web (autor de una página web, licencia, etc.), siendo más específicos está dirigido a la representación de los metadatos. En otras palabras, define la sintaxis y modelos de datos para la representación semántica de los datos. RDF se basa en los estándares de URIs y Unicode aparte de que también se puede presentar en XML.

El principal elemento de construcción en RDF es la tripleta o sentencia, que consiste en dos nodos (sujeto y objeto) unidos por un predicado, donde los nodos representan recursos, y los arcos propiedades. Por ejemplo una sentencia podría expresar el hecho de que el ajedrecista (predicado) de una partida de ajedrez (sujeto) fue la persona Bobby Fischer (objeto). Encadenando estas tripletas se construyen grafos o redes semánticas para la Web. [2]

Entonces RDF nos permitirá escribir "metainformación" para cualquier tipo de datos y trabajar con ellos. Podremos describir las propiedades de los documentos y de cualquier recurso. Además nos permitirá la interoperabilidad entre diversas aplicaciones sin pérdida del significado de los datos

## II. DOTA 2

DOTA2 es un juego de estrategia de acción en tiempo real en la que 2 equipos de 5 jugadores combaten entre sí con el objetivo de destruir la base enemiga, para esto cada jugador contará con 1 héroe de 121 seleccionables que posee diferentes características y habilidades.

**Héroes:** Los héroes son las unidades controladas por el jugador con habilidades especiales y únicas. Los héroes tienen 3 atributos principales, como se puede apreciar en

la figura 1 en rojo tenemos el atributo de fuerza, en verde el atributo de agilidad y azul el atributo de inteligencia.

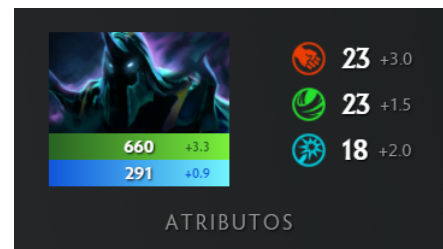


Figura 1. Heroe

**Habilidades:** El este trabajo gira en torno principalmente en torno las habilidades de los héroes, ya que existen una gran cantidad de estas y además son muy variadas, una clasificación básica de las habilidades puede ser esta:

- Active
- Passive
- Autocast

Dentro de las habilidades pasivas podemos también encontramos las auras que pueden ser positivas y negativas. Además, dentro de las habilidades activas estas se pueden clasificar de la siguiente manera:

- Target Unit
- Target Point Or Unit
- Target Area
- Vector Targeting
- NoTarget
- Toggle

Hemos considerado estas clasificaciones para crear la estructura RDF.

## III. ESTRUCTURA RDF

Los héroes están asociados con tres tipos de atributos: strength, agility, intelligence.

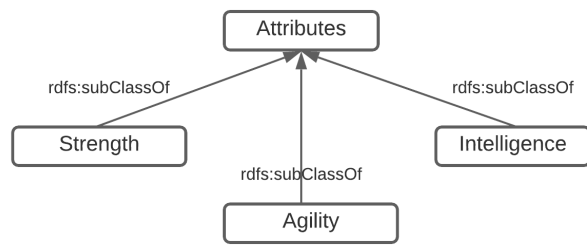


Figura 2. Atributos de un héroe.

Los héroes poseen dos modos de ataque : melee y ranged.

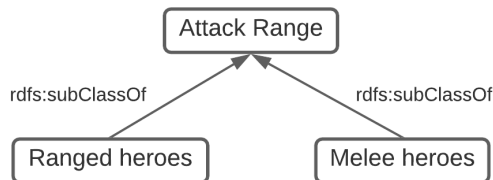


Figura 3. Modos de ataque.

Los héroes poseen tres tipos de habilidades : active , passive y autocast.

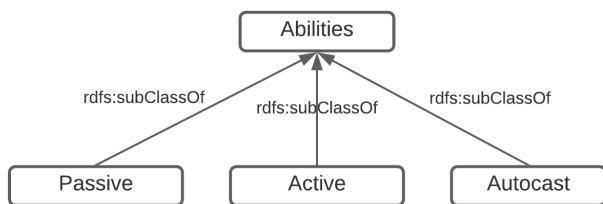


Figura 4. Tipos de habilidades.

Las habilidades activas se dirigen de diferentes maneras : target unit, target point or unit , target point , no target.

La habilidad target point requieren de target area o vector targeting.

Las habilidad toggle es siempre no target.

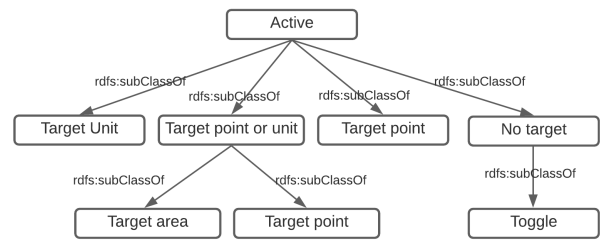


Figura 5. Tipos de habilidades activas.

Las auras son habilidades passive. Existen dos tipos de auras : positive aure y negative aure

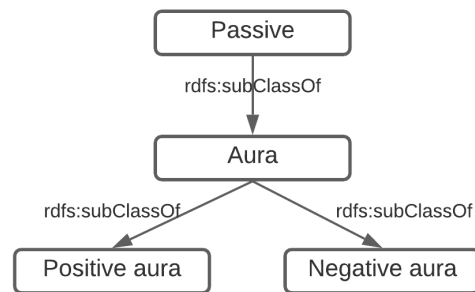


Figura 6. Tipos de habilidades pasivas.

#### IV. DESCRIPCIÓN DE RECURSOS USANDO RDF

Ahora nos encontramos con que la base del modelo RDF es un triplete con tres nodos: un sujeto (el recurso) tiene un predicado (propiedad) con un objeto determinado (valor o literal). Eso se podría mostrar en forma de gráfico entre nodos y flechas. Los nodos representan los recursos. Las flechas representan propiedades de los nodos y representan cadenas de literales.

```

g.add((razor, name, Literal("Razor")))
g.add((razor, RDF.type, n.agility))
g.add((razor, RDF.type, rangedHeroes))
g.add((razor, abilities, active))
g.add((razor, abilities, passive))
g.add((razor, agi, Literal(22)))
  
```

Figura 7. Descripción del héroe razor.

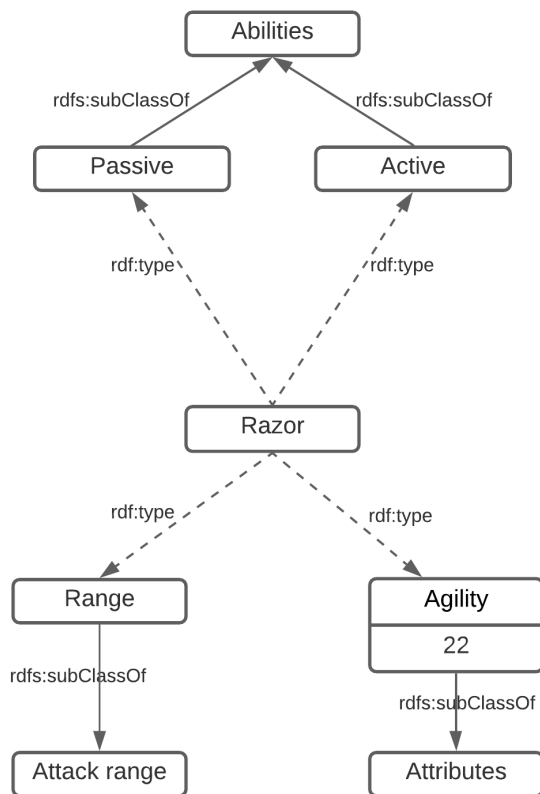


Figura 8. Ejemplo descripción de recursos .

## V. MANIPULACIÓN DE RDF

Encontrar los tipos asociados a la instancia: arcWarden

```
for s in g.transitive_objects(arcWarden, RDF.type):
    print(g.value(s, name))
```

Arc\_Warden  
Ranged\_Heroes  
Agility

Figura 9. Inferencia 1.

Encontrar las dependencias para toggle

```
for s in g.transitive_objects(toggle, RDFS.subClassOf):
    print(g.value(s, name))
```

Toggle  
No target  
Active  
Abilities

Figura 10. Inferencia 2.

Encontrar las dependencias para auras negativas

```
for s in g.transitive_objects(negativeAura, RDFS.subClassOf):
    print(g.value(s, name))
```

Negative aura  
Aura  
Passive  
Abilities

Figura 11. Inferencia 3.

Todos los heroes de agilidad

```
for t in g.subjects(None, n.agility):
    print(g.value(t, name))
```

Phantom\_Assassin  
Bloodseeker  
Clinkz  
Sniper  
Riki  
Ursa  
Phantom\_Lancer  
Ember\_Spirit  
Slark  
Monkey\_King  
Nyx\_Assassin  
Arc\_Warden  
Medusa  
Anti-Mage  
Troll\_Warlord  
Weaver  
Venomancer  
Viper  
Drow\_Ranger  
Meepo  
Gyrocopter  
Mirana  
Templar\_Assassin  
Shadow\_Fiend  
Vengeful\_Spirit  
Razor  
Broodmother  
Terrorblade  
Lone\_Druid  
Hoodwink  
Juggernaut  
Luna  
Morphling  
Spectre  
Pangolier  
Faceless\_Void  
Bounty\_Hunter  
Naga\_Siren

Figura 12. Inferencia 4.

Nombre y atributos de un héroe

```
def estadistica(nombre):
    flag = True
    for s, p, o in g.triples((None, None, None)):
        if g.value(s, name) == literal(nombre) and flag:
            print('Nombre:' + g.value(s, name))
            print('Fuerza:' + g.value(s, str))
            print('Agilidad:' + g.value(s, agi))
            print('Inteligencia:' + g.value(s, int))
            flag = False
```

estadistica('Abaddon')

Nombre:Abaddon  
Fuerza:23  
Agilidad:23  
Inteligencia:18

Figura 13. Inferencia 5.

## VI. CONCLUSIONES

Se logró crear una estructura RDF basada en el juego dota 2, aunque el juego ofrece un gran libertad al crear un estructura RDF debido a su gran cantidad de recursos, nos hemos centrando principalmente en los recursos sobre las mecánicas del juego respecto a las habilidades. Así mismo también hemos tomado en cuenta la estadísticas básicas de los héroes. La estructura creada también ha sido validada por el servicio de validación RDF.

## REFERENCIAS

- [1] <https://nextweb.gnoss.com/recurso/rdf-resource-description-framework-concepto/8934f06f-9bd5-49a7-92f6-49cadf9ce789>
- [2] <https://blog.infotics.es/2014/04/15/como-citar-una-url-con-bibtex/>
- [3] <https://github.com/JuniorGT/IA-PC1-GRUPO11>
- [4] ALLEMANG,D.HENDLER J.(2011).Semantic web for the working ontologist: effective modeling in RDFS and OWL (2nd edition).Massachusetts.Capítulo 3. RDF-The basis of the Semantic Web.