

## **Relazione di progetto**

### **1. Motivazioni**

Il mercato dei videogames è, ad oggi, uno dei più grandi e redditizi di sempre, con investimenti e ricavi che superano i miliardi di dollari ogni anno e anche il 2020 non ha fatto eccezione, con una crescita, ancora una volta, esponenziale. Sviluppare un gioco richiede ormai una quantità di risorse notevoli, con budget equiparabili, se non superiori, a quelli del cinema. È richiesto infatti uno staff tecnico altamente specializzato con competenze che spaziano dall'informatica fino all'arte e che usa le tecnologie più recenti e all'avanguardia al fine di produrre il miglior gioco possibile. Sebbene i ricavi possano risultare notevoli (ci sono giochi che hanno sfiorato i 6 miliardi di dollari) sono sempre di più le software house che, in seguito ad un solo gioco che non ha venduto quanto previsto, rischiano la chiusura o il ridimensionamento. In questo scenario, ha assunto notevole importanza il risvolto che un gioco ottiene dalla critica e dai videogiocatori. Un gioco che è stato accolto bene dalla critica si è dimostrato, infatti, molto spesso essere un successo anche di vendite e di pubblico, rendendo così fondamentali i siti specializzati nelle recensioni di un videogame.

### **2. Requirements**

Si vuole realizzare una piattaforma destinata ai videogiocatori ed esperti del settore (es giornalisti, sviluppatori, publisher/finanziatori) con l'obiettivo di fornire una panoramica generale sui videogames riguardante i risultati ottenuti da ciascun gioco. In particolar modo si vuole evidenziare il successo riscosso da un gioco sia in termini di recensioni che di premi vinti. Gli utenti della piattaforma, infatti, possono rilasciare delle recensioni con un voto per uno specifico gioco, esprimendo il gradimento con un voto da 1 a 10. Tra gli utenti troviamo sia gli appassionati, sia i recensori (soprannominati esperti), che lavorano per una redazione di una qualche testata giornalistica. I giochi particolarmente meritevoli hanno anche dei premi che vengono assegnati dalle varie redazioni, solitamente alla fine dell'anno.

La piattaforma è utile sia per gli appassionati, che possono esprimere il loro feedback circa la qualità di un gioco e confrontarsi con il parere degli altri utenti, sia per gli sviluppatori, che hanno la possibilità di analizzare e comprendere meglio i fattori che hanno determinato il

successo (o insuccesso) del proprio prodotto. Un altro riscontro particolarmente importante è la possibilità di guidare un utente nell'acquisto di un videogioco, indirizzandolo verso i migliori titoli disponibili sul mercato.

### **3. Descrizione dominio**

Gli elementi fondamentali del dominio sono:

- 1) il videogioco, caratterizzato dal titolo, data di uscita, genere e piattaforme su cui è disponibile;
- 2) la recensione, ossia un'analisi sui pregi e difetti di un gioco rilasciate da un videogiocatore oppure da un redattore, caratterizzata da un voto che va da 1 a 10 che riassume la qualità complessiva del gioco;
- 3) La redazione e la testata giornalistica presso cui lavorano i redattori, che hanno il compito di recensire videogioco
- 4) Il premio, assegnato da una redazione per conto di una testata giornalistica ad uno specifico gioco, il quale rappresenta un particolare riconoscimento che si è voluto attribuire al gioco.

Fonti bibliografiche:

<https://www.metacritic.com>

[https://it.wikipedia.org/wiki/Testata\\_giornalistica](https://it.wikipedia.org/wiki/Testata_giornalistica)

<https://thegameawards.com>

Vari siti specializzati nelle recensioni di videogiochi come:

<https://it.ign.com>

<https://www.spaziogames.it>

<https://multiplayer.it>

## 4. Documentazione

Per la progettazione dell'ontologia ci siamo basati sulle seguenti risorse. In particolare, la prima immagine mostra la scheda riassuntiva di un gioco, con riferimento al suo genere, alle piattaforme in cui è uscito, le recensioni, i premi e i voti che ha ottenuto sia da parte della critica che dell'utenza. Più nel dettaglio, nei riquadri in rosso si trovano le informazioni ritenute di rilievo per l'ontologia:

- 1) Titolo del videogioco con piattaforma, data di uscita e publisher
- 2) Voto medio della critica
- 3) Voto medio degli utenti appassionati
- 4) Ulteriori informazioni sul gioco come genere e sviluppatore
- 5) Premi che ha vinto il gioco
- 6) Le recensioni rilasciate dalla critica e dagli utenti

**THE LAST OF US PART II** PlayStation 4  
SCEA, Sony Interactive Entertainment | Release Date: Jun 19, 2020

**Summary** **Critic Reviews** **User Reviews** **Details & Credits** **Trailers & Videos**

**Metascore**  
Universal acclaim  
based on 121 Critic Reviews  
**93**  
What's this?

**User Score**  
Mixed or average reviews  
based on 151273 Ratings  
**5.7**  
Your Score: 0

**Developer:** Naughty Dog  
**Genre(s):** General, Action Adventure, Survival  
**# of players:** No Online Multiplayer  
**Cheats:** On GameFAQs  
**Rating:** M  
**More Details and Credits**

**CRITIC REVIEWS**  
Positive: 113  
Mixed: 8  
Negative: 0  
**100** Trusted Reviews  
Jun 12, 2020  
The Last of Us Part II is a masterful swansong for the PlayStation 4, and arguably the finest, most accomplished project Naughty Dog has ever embarked on.  
All this publication's reviews | Read full review

**USER REVIEWS**  
Positive: 36,897  
Mixed: 4,294  
Negative: 35,244  
**10** soWokeUasleep  
Jun 25, 2020  
The game looks amazing, improving on even the last of us remastered and looks better than Naughty Dog's uncharted 4.  
The game play is more... Expand  
1 of 4 users found this helpful

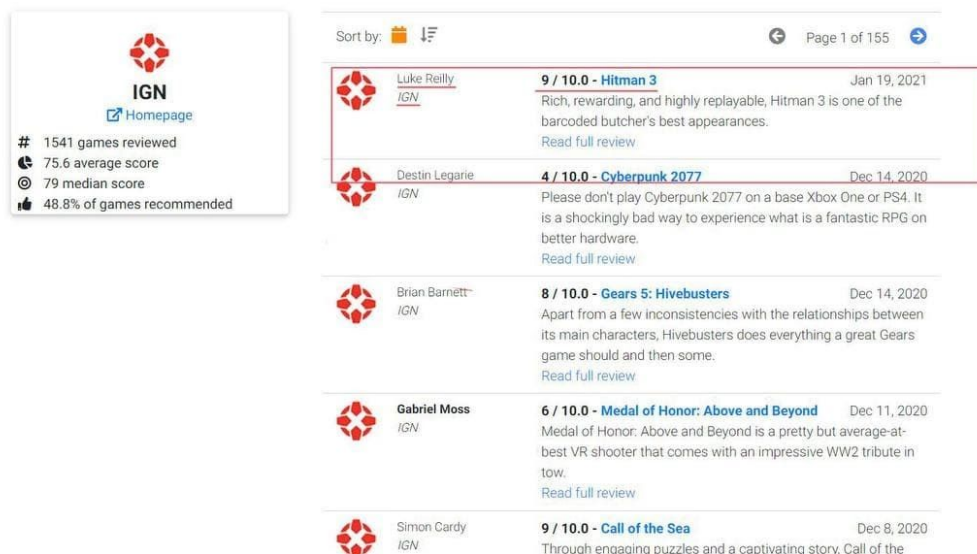
**AWARDS & RANKINGS**  
#2 Best PS4 Game of 2020  
#1 Most Discussed PS4 Game of 2020  
#2 Most Shared PS4 Game of 2020

**ESSENTIAL LINKS**  
New PS4 Releases by Date  
New PS4 Releases by Title  
Best Recent Releases for PS4  
Upcoming PS4 Release Calendar  
2021 High Scores - PS4  
All-Time High Scores - PS4  
All PS4 Games: A-Z Index  
Best Games of 2020  
Release Dates for Notable Upcoming Games  
PS Now, PS Plus, and Other New Free PS4 Games  
More articles

**CURRENT PS4 GAMES**  
Full List

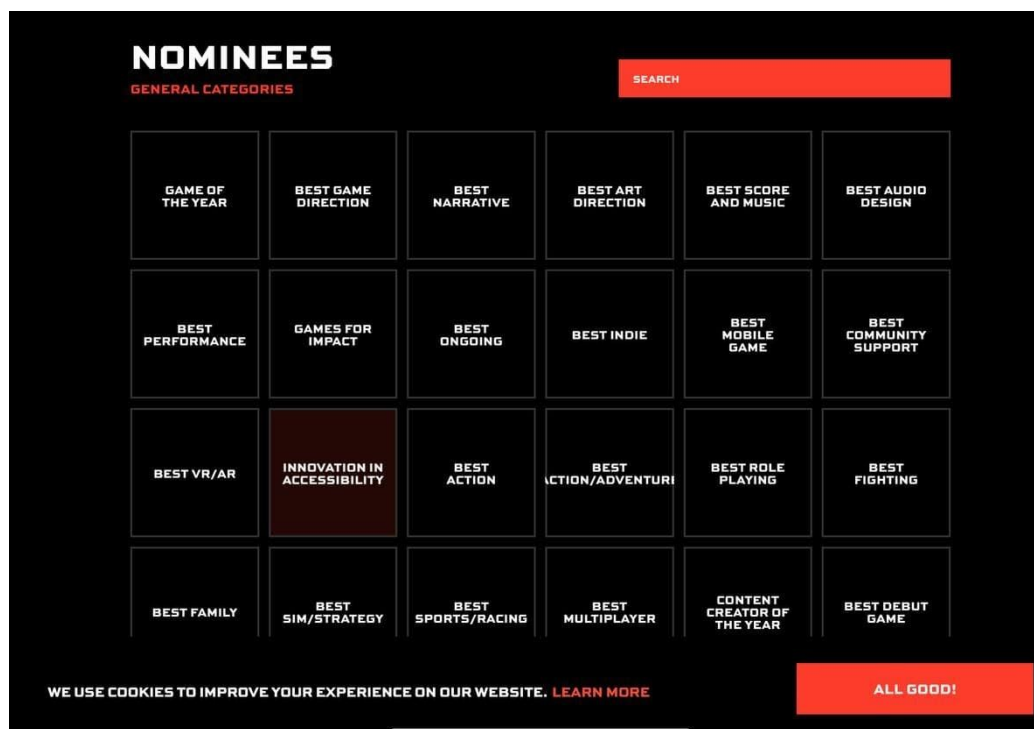
<https://www.metacritic.com/game/playstation-4/the-last-of-us-part-ii>

La seconda immagine, invece, è stata utile per modellare meglio le recensioni, includendo anche il redattore e la testata per cui lavora.



<https://opencritic.com/outlet/56/ign>

Infine la terza immagine riguarda più nel dettaglio i premi che vengono assegnati annualmente ai giochi più meritevoli. Sulla base di questa risorsa sono state individuate due categorie di premi, chiamate specialAward (che comprende premi solitamente molto ambiti come il game of the year) e categoryAward (che comprende premi sì importanti ma meno decisivi come ad esempio il best family), solitamente legati ad una categoria particolare come il genere.



### **Esempio di A-box**

Viene qui riportato un esempio tratto dall' ontologia videogamesAwards2020 da noi realizzata. L'esempio è riferito alla risorsa :videogioco1 ovvero il videogame intitolato "Sekiro: Shadows Die Twice" , rilasciato il 22 Marzo 2019 per le seguenti piattaforme: Playstation 4, google stadia e hp omen 17. Questo titolo ha riscosso parecchio successo, ottenendo il premio speciale di *game of the year*, rilasciato dalla testata giornalistica *Eurogamer*, e 3 premi per la categoria *Action* che hanno permesso al gioco di far parte dei giochi *MustHave* ovvero quelli fortemente suggeriti dalla piattaforma. Tale successo si può riscontrare anche nelle recensioni, che risultano positive sia da parte della critica (con un voto medio di 9.5) sia dagli utenti con un voto medio di 8.8. I premi sopra citati possono essere visualizzati attraverso l'insieme denominato awardSet1, descritto formalmente con l'uso del pattern ODP "Set". E' possibile anche risalire alla redazione e quindi alla testata giornalistica che si è occupata di una qualche recensione del videogioco. Ad esempio l'utente Aldo, che lavora come recensore per la redazione della testata giornalistica IGN, ha rilasciato una recensione positiva con voto 9 (in questo caso è stata utilizzata una variante del pattern role).

### **Allineamento con ontologie fondazionali**

Per quanto riguarda l'ordinamento ontologico, sono state utilizzate due ontologie: Provenance e Schema.org.

La prima ontologia è stata utilizzata per descrivere le informazioni riguardanti la provenienza delle reviews. Ad esempio, la review è stata definita come sottoclasse di prov:Entity mentre l'utente, ossia colui che scrive la recensione è sottoclasse di prov:Person. L'utente (Person, che a sua volta è sottoclasse di prov:Agent) e la recensione (Entity) sono legate dalla property :wasAttributedTo, sovra-property del nostro :makeReview.

La scelta di adottare Schema.org è risultata dal fatto che l'ontologia è attualmente una tra le più utilizzate complete nel web semantico. Utilizzando schema è stato possibile allineare numerose nostre classi, come il videogioco, l'utente, la recensione dell'utente (UserReview) e quella del recensore (CriticReview, da noi chiamata ExpertReview), il newsPaper (testata giornalistica).

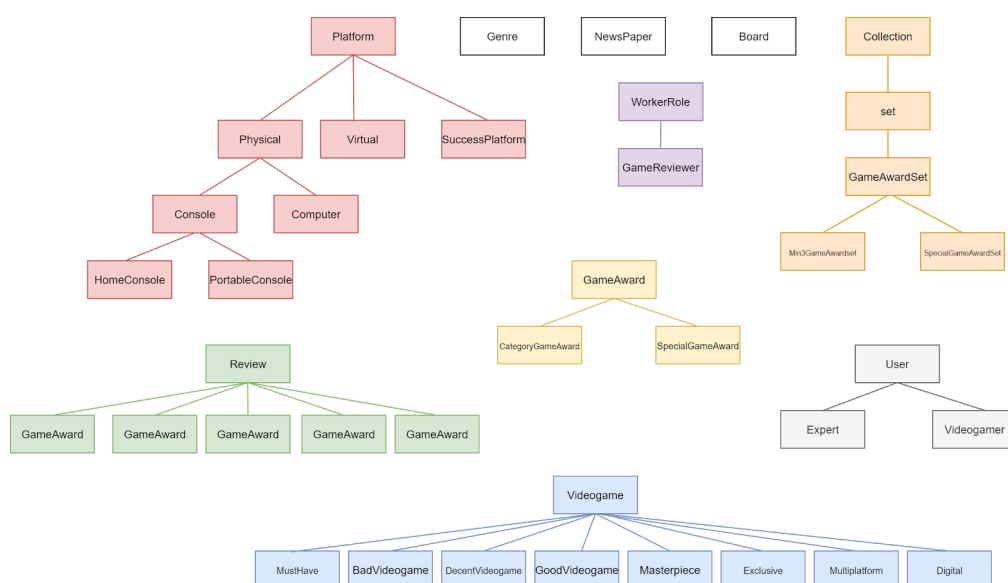


## 5. LODE

È allegata insieme alla relazione la documentazione dell'onologia realizzata attraverso LODE.

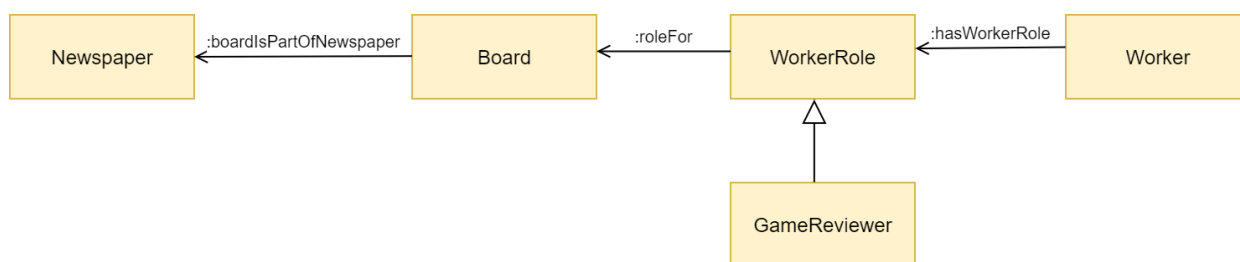
## 6. Visualizzazione

Viene qui riportata la tassonomia delle classi. Per semplicità è stato ommesso owl:Thing, ma tutte le classi che non hanno una superclasse sono sottoclassi di Thing. È possibile individuare 10 classi “core”, che sono rispettivamente: Platform, Genre, NewsPaper, Board, Collection, Review, GameAward, User, Videogame e WorkerRole.



### Principali template e knowledge graph

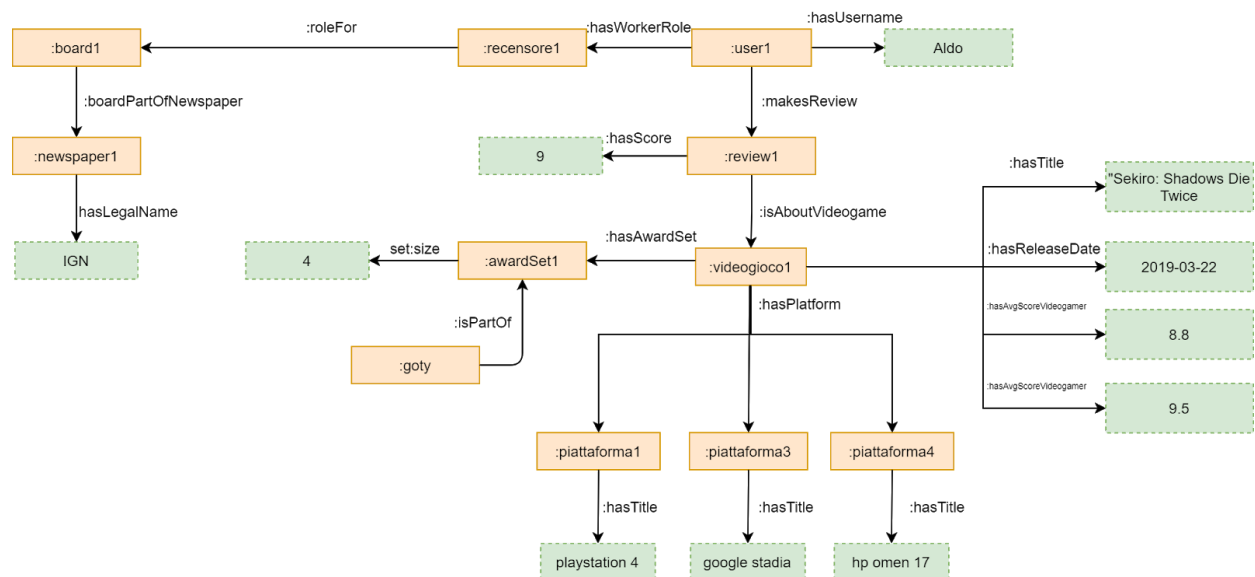
Tra i principali template realizzati si possono trovare in primo luogo il pattern Set e Role, utilizzati rispettivamente per esprimere l'appartenenza di un premio ad un insieme di premi assegnato poi ad un videogioco e per assegnare il ruolo di recensore ad un utente. Il pattern Role è stato parzialmente riadattato ed è riassumibile in questo modo:



Una redazione (board), che fa parte di una testata giornalistica (newspaper) solitamente ha al suo interno una o più persone che vi lavorano e che svolgono ruoli diversi. Sulla base di questo, il pattern role è stato riadattato. In particolare vi è un worker che ha uno specifico ruolo per una specifica redazione che fa a sua volta parte di un giornale. Il worker di fatto coincide con l'utente esperto della piattaforma. Segue la tabella relativa all'esempio proposto prima e il knowledge graph.

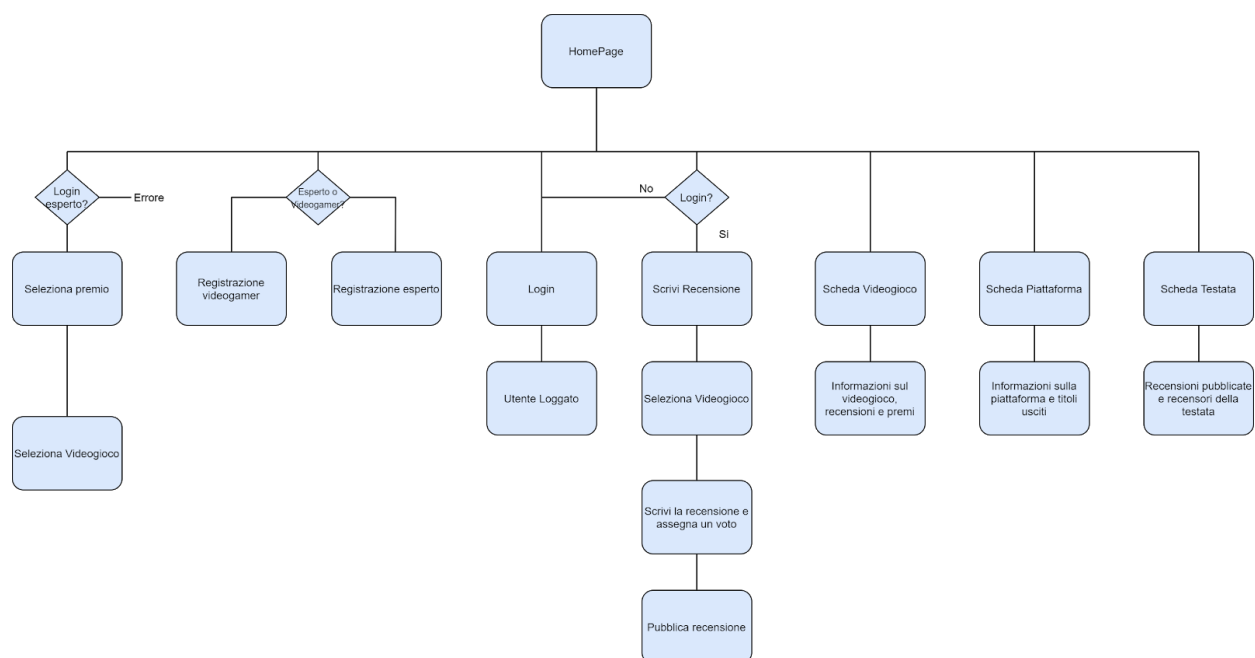
Soggetto	Predicato	Oggetto
:user1	:hasUsername	"Aldo"
:user1	:hasWorkerRole	:recensore1
:recensore1	:roleFor	:board1
:board1	:boardIsPartOfNewspaper	:newspaper1
:newspaper1	:hasLegalName	"IGN"
:user1	:makesReview	:review1
:review1	:hasScore	9
:review1	:isAboutVideogame	:videogioco1
:videogioco1	:hasTitle	"Sekiro: Shadows Die Twice"
:videogioco1	:hasReleaseDate	"2019-03-22"
:videogioco1	:hasAverageRatingVideogamer	8.8
:videogioco1	:hasAverageRatingExpert	9.5
:videogioco1	:hasAwardSet	:awardSet1
:awardSet	:size	4
:goty	:isPartOf	:awardSet1
:videogioco1	:hasPlatform	:piattaforma1
:videogioco1	:hasPlatform	:piattaforma3
:videogioco1	:hasPlatform	:piattaforma4
:piattaforma1	:hasTitle	"Playstation 4"
:piattaforma3	:hasTitle	"google stadia"
:piattaforma4	:hasTitle	"Hp Omen 17"





## 7. Queries Sparql

### Flow-chart e interazione



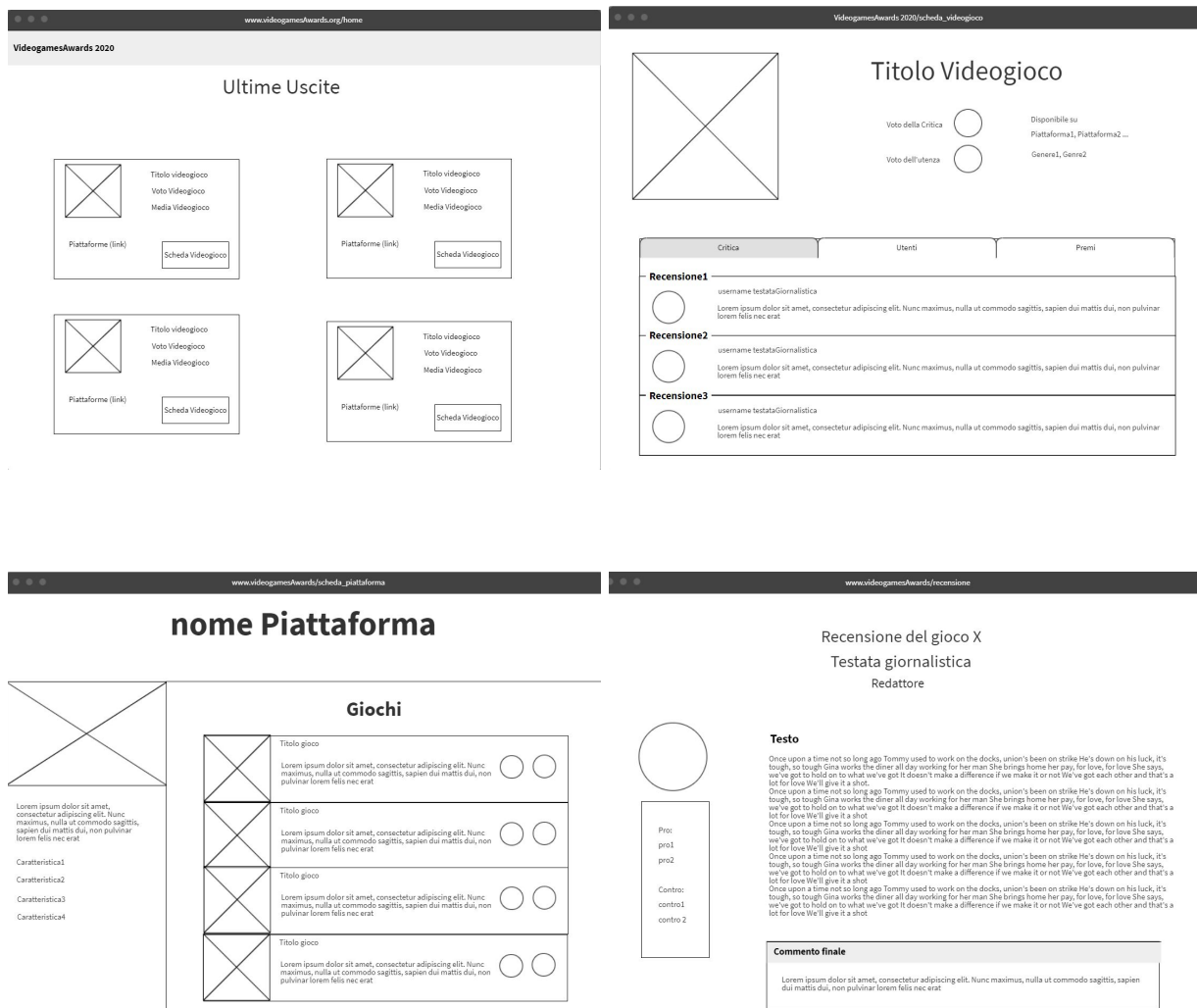
Il diagramma di flusso mostra una visione ad alto livello dell'interazione tra la piattaforma e l'utente. Il punto di partenza è rappresentato dalla pagina homepage (Home nel flowchart), che contiene una panoramica generale della piattaforma, come mostrato nei mock up successivi. Dalla home è possibile accedere a qualsiasi pagina (ad esempio cliccando su uno

specifico videogioco si andrà nella sua pagina dedicata, avente tutte le caratteristiche del gioco, le piattaforme su cui è uscito, le recensioni e i premi vinti).

## Mock Up

Seguono ora i principali mockups realizzati come supporto per l'implementazione del client.

Più nello specifico vengono mostrati i mockups relativi alla homepage, alla scheda del videogioco, della piattaforma e di una recensione.



## Esempio con dati reali

Un esempio con dati reali può essere il seguente, basato sul client. In ordine vengono mostrate le 4 schermate dei mockup, implementate in Angular.

VideogamesAwards2020

Ultime uscite

Titolo	Uscita	Genere	Piattaforma	Voto Critica	Voto Utenti	Recensioni	Premi Vinti
BioShock	2020-11-13T11:21:42+00:00	tps	PlayStation 4	8.5	8.4	1	1
Dishonored 2	2018-04-20T11:21:42+00:00	action/gdr	PlayStation 4,Google Stadia,Hp Omen 17	8.8	9.5	3	1
Helblade: Senais Sacrifice	2008-10-14T00:00:00	tps	Xbox 360,PlayStation 3	8.8	7.7	1	0
Kingdom Come: Deliverance	2017-07-21T00:00:00	tps	Xbox 360,PlayStation 3	8.6	5.3	1	0
Mass Effect 2	2010-06-27T11:21:42+00:00	tps	PlayStation 4	8.3	8.5	1	0
Sekiro: Shadows Die Twice	2019-03-22T11:21:42+00:00	action	PlayStation 4,Google Stadia,Hp Omen 17	9.5	8.8	4	4
The Last of Us	2011-10-	odr	PlayStation 4,Nintendo	7.5	9.5	1	1

VideogamesAwards2020

Sekiro: Shadows Die Twice

Publishers

Activision  
From Software

Director

Hidetaka Miyazaki

Origin

Giappone

Genre

action

Platforms

Playstation 4  
Google Stadia  
Hp Omen 17

Average Expert Score

9.5

Average Videogamer Score

8.8

Awards Won

3

Recensioni della Critica

VideogamesAwards2020

PlayStation 4

Nome

PlayStation 4

Prodotta da

Sony Interactive Entertainment

Unità Vendute

11380000

Giochi Usciti

Videogame	Voto Critica	Voto Utenti
Sekiro: Shadows Die Twice	9.5	8.8
Dishonored 2	8.8	9.5
BioShock	8.5	8.4
Mass Effect 2	8.3	8.5

VideogamesAwards2020

Review

Testata

Spaziogames

Redattore

Gianfranco

Testo

Once upon a time not so long ago Tommy used to work on the docks, until's been on strike He's down on his luck, it's tough, so tough Gina works the diner all day working for her man She brings home her pay, for love, for love She says, we've got to hold on to what we've got it doesn't make a difference if we make it or not We've got each other and that's a lot for love We'll give it a shot

Voto

9

Tipologia 1

ExpertReview

Tipologia 2

PositiveReview

VIDEOGAMES AWARDS 2020

>Lorem ipsum dolor sit amet consectetur, adipiscing elit. Iste atque ea quis molestias. Fugiat pariatur maxime quis culpa corporis vitae repudiandae aliquam voluptatem veniam, est atque cumque eum delectus sint!

## Query 1

```

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX vg: <http://www.videogamesAwards.org/videogamesAwards2020#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select ?videogame ?videogameTitle
where{
    ?videogame rdf:type vg:Videogame.
    ?videogame vg:hasTitle ?videogameTitle.
    ?videogame rdf:type vg:MustHave.
    ?videogame vg:hasReleaseDate ?releaseDate.
}
ORDER BY DESC(?releaseDate)

```

Seleziona i videogiochi con relativo titolo, i quali sono classificati come giochi *MustHave* ordinandoli in base alla data di uscita. I giochi *MustHave* sono quelli che hanno ricevuto un premio speciale oppure 3 premi di categoria.

### Risultato:

	videogameTitle	videogameTitle
1	<a href="#">videogamesAwards2020:videogioco1</a>	"Sekiro: Shadows Die Twice"

2	<a href="#">videogamesAwards2020:videogioco2</a>	"Dishonored 2"
3	<a href="#">videogamesAwards2020:videogioco5</a>	"The Last of Us"

## Query 2

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX vg: <http://www.videogamesAwards.org/videogamesAwards2020#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

select ?newspaper ?legalName
where {
    ?newspaper rdf:type vg:Newspaper.
    ?newspaper vg:legalName ?legalName.
}
```

Seleziona tutte le testate giornalistiche registrate alla piattaforma con il relativo nome legale.

## Risultato:

	newspaper	legalName
1	<a href="#">videogamesAwards2020:newspaper1</a>	"Spaziogames"
2	<a href="#">videogamesAwards2020:newspaper2</a>	"Multiplayer"
3	<a href="#">videogamesAwards2020:newspaper3</a>	"IGN"
4	<a href="#">videogamesAwards2020:newspaper4</a>	"Eurogamer"

## Query 3

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vg: <http://www.videogamesAwards.org/videogamesAwards2020#>

SELECT ?board ?worker ?username ?workerRole
WHERE{
    ?board rdf:type vg:Board.
    ?board vg:boardIsPartOfNewspaper [].
    ?worker vg:hasWorkerRole ?workerRole.
    ?worker vg:hasUsername ?username.
}
```

```

    ?workerRole vg:roleFor ?board.
}
ORDER BY ?username

```

Seleziona le redazioni registrate alla piattaforma elencando lo staff che lavora per esse. Per ogni lavoratore viene selezionato il suo username, usato al momento della registrazione nella piattaforma, e il ruolo da lui ricoperto all'interno della redazione.

**Risultato:**

	board	worker	username	workerRole
1	<a href="#">videogamesAwards2020:board1</a>	<a href="#">videogamesAwards2020:user1</a>	"Aldo"	<a href="#">videogamesAwards2020:recensore1</a>
2	<a href="#">videogamesAwards2020:board3</a>	<a href="#">videogamesAwards2020:user3</a>	"Giacomo"	<a href="#">videogamesAwards2020:recensore3</a>
3	<a href="#">videogamesAwards2020:board1</a>	<a href="#">videogamesAwards2020:worker_1</a>	"Gianfranco"	<a href="#">Q613354:recensore_worker</a>
4	<a href="#">videogamesAwards2020:board2</a>	<a href="#">videogamesAwards2020:user2</a>	"Giovanni"	<a href="#">videogamesAwards2020:recensore2</a>
5	<a href="#">videogamesAwards2020:board4</a>	<a href="#">videogamesAwards2020:user4</a>	"Giuseppe"	<a href="#">videogamesAwards2020:recensore4</a>

## Query 4

```

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vg: <http://www.videogamesAwards.org/videogamesAwards2020#>

SELECT ?videogame ?title ?releaseDate ?platform ?platformName ?genre
?awardsNumber ?reviewsNumber ?scoreE ?scoreU
WHERE
{
  {
    ?videogame vg:hasTitle ?title;
              vg:hasReleaseDate ?releaseDate;
              vg:hasPlatform ?platform;
              vg:hasGenre ?genre;
              vg:hasAverageRatingExpert ?scoreE;
              vg:hasAverageRatingVideogamer ?scoreU.
    ?platform vg:platformName ?platformName.
  }
}

```

```

    }
    {
        SELECT ?videogame (COUNT(?award) AS ?awardsNumber)
        WHERE{
            ?videogame vg:videoGameHasAward ?award.
        }
        GROUP BY ?videogame
    }

    {
        SELECT ?videogame (COUNT(?review) AS ?reviewsNumber)
        WHERE{
            ?videogame vg:videoGameHasReview ?review.
        }
        GROUP BY ?videogame
    }
}

```

Seleziona le informazioni riguardante un videogame quali: titolo, data di rilascio, piattaforme sulla quale è possibile giocare il titolo, generi, numero di premi ricevuti, numero di recensioni totali per il gioco, voto medio delle recensioni rilasciate dalla critica(?scoreE) e dagli appassionati(?scoreU). Il voto medio selezionato in questa query è quello che viene inserito a mano per ogni videogioco (non viene cioè calcolato on-the-fly).

**Risultato:** (La tabella è stata semplificata per migliorarne la lettura)

	title	releaseDate	platformName	genreLabel	awardsNumber	reviewsNumber	scoreE	scoreU
1	"Sekiro: Shadows Die Twice"	"2019-03-22T11:21:42+00:00" ^^xsd:dateTime	"PlayStation 4"	"action" ^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"4"^^xsd:integer	"4"^^xsd:integer	"9.5"^^xsd:double	"8.8"^^xsd:double
2	"Sekiro: Shadows Die Twice"	"2019-03-22T11:21:42+00:00" ^^xsd:dateTime	"Google Stadia"	"action" ^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"4"^^xsd:integer	"4"^^xsd:integer	"9.5"^^xsd:double	"8.8"^^xsd:double
3	"Sekiro: Shadows Die Twice"	"2019-03-22T11:21:42+00:00" ^^xsd:dateTime	"Hp Omen 17"	"action" ^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"4"^^xsd:integer	"4"^^xsd:integer	"9.5"^^xsd:double	"8.8"^^xsd:double
4	"Dishonored 2"	"2018-04-20T11:21:42+00:00" ^^xsd:dateTime	"PlayStation 4"	"action" ^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"1"^^xsd:integer	"3"^^xsd:integer	"8.8"^^xsd:double	"9.5"^^xsd:double

5	"Dishonored 2"	"2018-04-20T11:21:42+00:00" ^^xsd:dateTime	"Google Stadia"	"action" ^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"1"^^xsd:integer	"3"^^xsd:integer	"8.8"^^xsd:double	"9.5"^^xsd:double
6	"Dishonored 2"	"2018-04-20T11:21:42+00:00" ^^xsd:dateTime	"Hp Omen 17"	"action" ^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"1"^^xsd:integer	"3"^^xsd:integer	"8.8"^^xsd:double	"9.5"^^xsd:double
7	"Dishonored 2"	"2018-04-20T11:21:42+00:00" ^^xsd:dateTime	"PlayStation 4"	"gdr"^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"1"^^xsd:integer	"3"^^xsd:integer	"8.8"^^xsd:double	"9.5"^^xsd:double
8	"Dishonored 2"	"2018-04-20T11:21:42+00:00" ^^xsd:dateTime	"Google Stadia"	"gdr"^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"1"^^xsd:integer	"3"^^xsd:integer	"8.8"^^xsd:double	"9.5"^^xsd:double
9	"Dishonored 2"	"2018-04-20T11:21:42+00:00" ^^xsd:dateTime	"Hp Omen 17"	"gdr"^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"1"^^xsd:integer	"3"^^xsd:integer	"8.8"^^xsd:double	"9.5"^^xsd:double
10	"Bioshock"	"2020-11-13T11:21:42+00:00" ^^xsd:dateTime	"PlayStation 4"	"fps"^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"1"^^xsd:integer	"1"^^xsd:integer	"8.5"^^xsd:double	"8.4"^^xsd:double
11	"The Last of Us"	"2011-10-06T11:21:42+00:00" ^^xsd:dateTime	"PlayStation 4"	"gdr"^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"1"^^xsd:integer	"1"^^xsd:integer	"7.5"^^xsd:double	"9.5"^^xsd:double
12	"The Last of Us"	"2011-10-06T11:21:42+00:00" ^^xsd:dateTime	"Nintendo Switch"	"gdr"^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"1"^^xsd:integer	"1"^^xsd:integer	"7.5"^^xsd:double	"9.5"^^xsd:double
13	"The Last of Us"	"2011-10-06T11:21:42+00:00"^^xsd:dateTime	"Hp Omen 17"	"gdr"^^<http://www.w3.org/2000/01/rdf-schema#Literal>	"1"^^xsd:integer	"1"^^xsd:integer	"7.5"^^xsd:double	"9.5"^^xsd:double

## Query 5

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vg: <http://www.videogamesAwards.org/videogamesAwards2020#>

SELECT ?videogame ?avgExpert ?avgVideogamer
WHERE
{
  {
    SELECT ?videogame (AVG (?scoreE) AS ?avgExpert)
    WHERE{
      ?videogame vg:videogameHasReview ?review.
      OPTIONAL{
        ?review rdf:type vg:ExpertReview.
        ?review vg:hasScore ?scoreE.}
    }
    GROUP BY ?videogame
  }
  {
    SELECT ?videogame (AVG (?scoreU) AS ?avgVideogamer)
    WHERE{
      ?videogame vg:videogameHasReview ?review2.
      OPTIONAL{
        ?review2 rdf:type vg:VideoGamerReview.
        ?review2 vg:hasScore ?scoreU.}
    }
    GROUP BY ?videogame
  }
}
```

Vengono selezionati i videogiochi con il relativo voto medio della critica e il numero di recensioni ricevute. Il voto medio viene calcolato on-the-fly calcolando la media dei voti delle recensioni esperte realmente ricevute.

**Risultato:**

	videogame	avgExpert	avgVideogamer
1	<a href="#">videogamesAwards2020:videogioco1</a>	"8.833333333333333333333333" <small>^^xsd:decimal</small>	"8" <small>^^xsd:decimal</small>
2	<a href="#">videogamesAwards2020:videogioco2</a>	"9.666666666666666666666667" <small>^^xsd:decimal</small>	"0.0" <small>^^xsd:double</small>



	eogioco2	^^xsd:decimal	
3	videogamesAwards2020:videogioco3	"7.5"^^xsd:decimal	"0.0"^^xsd:double
4	videogamesAwards2020:videogioco4	"0.0"^^xsd:double	"8.5"^^xsd:decimal
5	videogamesAwards2020:videogioco5	"0.0"^^xsd:double	"9.5"^^xsd:decimal
6	videogamesAwards2020:videogame_1	"9"^^xsd:decimal	"0.0"^^xsd:double
7	videogamesAwards2020:videogame_2	"9"^^xsd:decimal	"0.0"^^xsd:double

## Query 6

```

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vg: <http://www.videogamesAwards.org/videogamesAwards2020#>

SELECT ?platform ?platformName ?numberVideogames ?success
WHERE {
  {
    ?platform vg:platformName ?platformName.
    OPTIONAL {
      ?platform rdf:type ?success.
      FILTER(?success = vg:SuccessPlatform) .
    }
  }

  {
    SELECT ?platform (COUNT(?videogame) AS ?numberVideogames)
    WHERE {
      ?platform vg:isPlatformFor ?videogame.
    }
    GROUP BY ?platform
    HAVING(?numberVideogames >= 2)
  }
}

```

Seleziona le piattaforme di gioco mostrando per ognuna di esse le seguenti informazioni: nome della piattaforma, numero di videogiochi supportati, campo ?success che dice se la piattaforma è di successo o no (nel caso in cui la piattaforma non sia di successo il campo sarà vuoto). Si

ricorda che una piattaforma è di successo quando supporta almeno 3 giochi diversi classificati come *MustHave*.

**Risultato:**

	platform	platformName	numberVideogames	success
1	<a href="#">videogamesAwards2020:piattaforma1</a>	"PlayStation 4"	"5"^^xsd:integer	<a href="#">videogamesAwards2020:Success Platform</a>
2	<a href="#">videogamesAwards2020:piattaforma3</a>	"Google Stadia"	"2"^^xsd:integer	
3	<a href="#">videogamesAwards2020:piattaforma4</a>	"Hp Omen 17"	"3"^^xsd:integer	<a href="#">videogamesAwards2020:Success Platform</a>
4	<a href="#">videogamesAwards2020:piattaforma_1</a>	"Xbox 360"	"2"^^xsd:integer	
5	<a href="#">videogamesAwards2020:piattaforma_2</a>	"PlayStation 3"	"2"^^xsd:integer	

## Query 7

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX vg: <http://www.videogamesAwards.org/videogamesAwards2020#>

SELECT ?videogame ?title (COUNT(?award) AS ?numberAwards)
WHERE {
    ?videogame vg:videoGameHasAward ?award;
               vg:hasTitle ?title.
}
GROUP BY ?videogame ?title
ORDER BY DESC(?numberAwards)
LIMIT 1
```

Seleziona il videogioco con il numero di premi più alto mostrando il suo titolo. Questo è ottenuto ordinando i videogiocchi in ordine decrescente di numero di premi e limitando ad 1 la visualizzazione dei risultati.

**Risultato:**

	videogame	title	numberAwards
--	-----------	-------	--------------

1	<a href="#">videogamesAwards2020:videogioco1</a>	"Sekiro: Shadows Die Twice"	"4"^^xsd:integer
---	--	-----------------------------	------------------

## Query 8

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX bd: <http://www.bigdata.com/rdf#>
PREFIX vg: <http://www.videogamesAwards.org/videogamesAwards2020#>
PREFIX set: <http://www.ontologydesignpatterns.org/cp/owl/set.owl#>

SELECT ?videogame ?title ?legalName ?username ?review ?score
WHERE {
    ?videogame vg:reviewedByNewsPaper ?newspaper.
    ?videogame vg:hasTitle ?title.
    ?newspaper vg:newspaperContainsBoard ?board.
    ?newspaper vg:legalName ?legalName.
    ?user vg:worksForBoard ?board.
    ?user vg:makeReview ?review.
    ?user vg:hasUsername ?username.
    ?review vg:reviewIsAbout ?videogame.
    ?review vg:hasScore ?score.

    FILTER(?title = "Sekiro: Shadows Die Twice")
}
ORDER BY DESC(?score)

```

Visualizza tutte le recensioni rilasciate per un gioco specifico da esperti che lavorano per una redazione giornalistica, specificando il voto. Per l'esempio è stato selezionato il videogame dal titolo "Sekiro: Shadows Die Twice".

### Risultato:

	videogame	title	legalName	username	review
1	<a href="#">videogamesAwards2020:videogioco1</a>	"Sekiro: Shadows Die Twice"	"Spaziogames"	"Aldo"	<a href="#">videogamesAwards2020:review1</a>
2	<a href="#">videogamesAwards2020:videogioco1</a>	"Sekiro: Shadows Die Twice"	"IGN"	"Giacomo"	<a href="#">videogamesAwards2020:review7</a>
3	<a href="#">videogamesAwards2020:videogioco1</a>	"Sekiro: Shadows Die"	"Eurogamer"	"Giuseppe"	<a href="#">videogamesAwards2020:re</a>

		Twice"			view4
--	--	--------	--	--	-------

## Federata 1

```

SELECT DISTINCT ?publisher ?publisherLabel ?origin ?originLabel ?director
?directorLabel ?genre ?platform ?avgVideogamer ?size
WHERE{
  {
    ?myVideogame vg:hasTitle ?myTitle.
    FILTER(?myTitle = "Sekiro: Shadows Die Twice").
    ?myVideogame vg:hasGenre ?genre.
    ?myVideogame vg:hasPlatform ?platform.
    ?myVideogame vg:hasAverageRatingVideogamer ?avgVideogamer.
    ?myVideogame vg:hasAwardSet ?set.
    ?set set:size ?size.
  }
  SERVICE <https://query.wikidata.org/sparql>{
    SELECT DISTINCT ?publisher ?videogameLabel ?publisherLabel ?origin
?originLabel ?director ?directorLabel WHERE {
      ?videogame wdt:P31 wd:Q7889;
      wdt:P1476 ?title;
      wdt:P123 ?publisher;
      wdt:P495 ?origin;
      wdt:P57 ?director.
      FILTER(CONTAINS(?title, "Sekiro: Shadows Die Twice"))
      SERVICE wikibase:label { bd:serviceParam wikibase:language
"[AUTO_LANGUAGE],it". }
    }
  }
}

```

La query federata numero 1 visualizza informazioni complete su un videogioco specifico (nell'esempio qui riportato è stato usato nuovamente il gioco "Sekiro: Shadows Twice Die"). Le informazioni recuperate da wikidata sono le seguenti: publisher, paese di origine, direttore di progetto. Le restante informazioni invece sono già presenti nella nostra ontologia ( genere, piattaforme, titolo, premi vinti ecc...

**Risultato:** (La tabella è stata semplificata per migliorarne la lettura)

	publisherLabel	originLabel	directorLabel	genre	platform	avgVideogamer	size
1	"From Software"@it	"Giappone"@it	"Hidetaka Miyazaki"@it	<a href="#">videogamesAwarded2020:g</a>	<a href="#">videogamesAwarded2020:p</a>	"8.8" <sup>^</sup> <sub>xsd:doubl</sub>	"3" <sup>^</sup> <sub>xsd:integer</sub>

				enere_acion	iattaforma1		
2	"Activision"@it	"Giappone"@it	"Hidetaka Miyazaki"@it	videogamesAwards2020:genere_acion	videogamesAwards2020:piattaforma1	"8.8"^^xsd:double	"3"^^xsd:integer
3	"From Software"@it	"Giappone"@it	"Hidetaka Miyazaki"@it	videogamesAwards2020:genere_acion	videogamesAwards2020:piattaforma3	"8.8"^^xsd:double	"3"^^xsd:integer
4	"Activision"@it	"Giappone"@it	"Hidetaka Miyazaki"@it	videogamesAwards2020:genere_acion	videogamesAwards2020:piattaforma3	"8.8"^^xsd:double	"3"^^xsd:integer
5	"From Software"@it	"Giappone"@it	"Hidetaka Miyazaki"@it	videogamesAwards2020:genere_acion	videogamesAwards2020:piattaforma4	"8.8"^^xsd:double	"3"^^xsd:integer
6	"Activision"@it	"Giappone"@it	"Hidetaka Miyazaki"@it	videogamesAwards2020:genere_acion	videogamesAwards2020:piattaforma4	"8.8"^^xsd:double	"3"^^xsd:integer

## Federata 2

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX bd: <http://www.bigdata.com/rdf#>
PREFIX vg: <http://www.videogamesAwards.org/videogamesAwards2020#>
PREFIX set: <http://www.ontologydesignpatterns.org/cp/owl/set.owl#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT ?console ?consoleLabel ?buildBy ?buildByLabel ?unitSold
WHERE{
  {
    ?myPlatform vg:platformName ?myPlatformName.
    FILTER(?myPlatformName = "Playstation 4")
  }
}

```

```

SERVICE <https://query.wikidata.org/sparql>{
  SELECT ?console ?consoleLabel ?buildBy ?buildByLabel ?unitSold
  WHERE {
    ?console wdt:P31 wd:Q10929058;
      rdfs:label ?consoleLabel.
    FILTER((LANG(?consoleLabel)) = "en")
    FILTER(lcase(?consoleLabel) = "playstation 4"@en)
    ?console wdt:P176 ?buildBy.
    ?buildBy rdfs:label ?buildByLabel.
    FILTER((LANG(?buildByLabel)) = "en")
    ?console wdt:P2664 ?unitSold
  }
}

```

La query federata numero 2 visualizza le informazioni riguardanti la piattaforma di gioco “PlayStation 4”. In particolare viene mostrato: nome della piattaforma, costruttore e unità totali vendute. Di queste informazioni solo il nome viene preso dalla nostra ontologia.

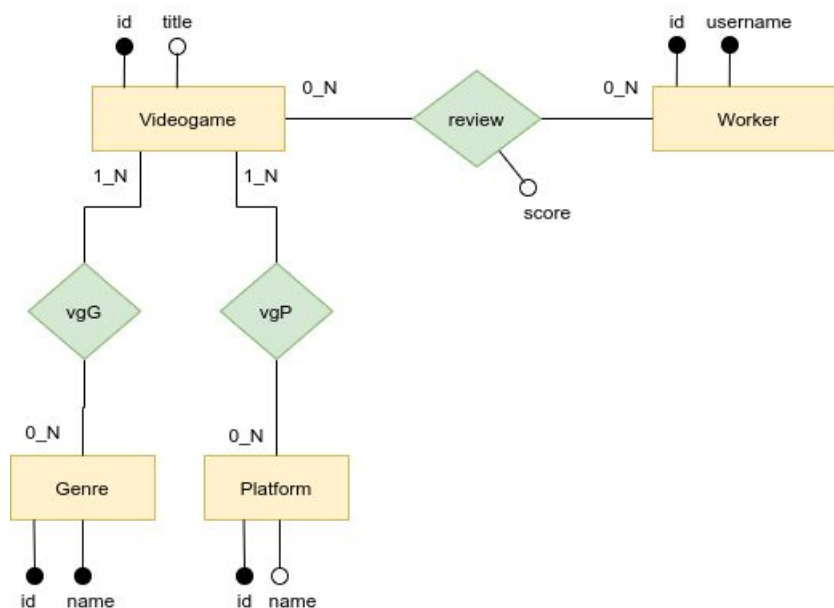
#### Risultato:

	console	consoleLabel	buildBy	buildByLabel	unitSold
1	<a href="http://www.wikidata.org/entity/Q5014725">http://www.wikidata.org/entity/Q5014725</a>	"PlayStation 4"@en	<a href="http://www.wikidata.org/entity/Q18594">http://www.wikidata.org/entity/Q18594</a>	"Sony Interactive Entertainment"@en	"113800000" ^^xsd:decimal

## 8. Importazione di dati nel grafo

Il mapping è stato realizzato per un sottoinsieme dell’ontologia videogamesAwards2020, prima dell’esportazione delle inferenze. In particolare è stato realizzato un database relazionale MYSQL per il salvataggio delle informazioni complete riguardante un videogioco e le recensioni rilasciate da un utente esperto (worker). Le informazioni salvate per un videogioco sono le seguenti: titolo, genere, piattaforma, data di rilascio, voto medio della critica e voto medio dei videogiocatori. Viene riportata anche una tabella per gli utenti esperti registrati alla piattaforma, ovvero coloro che lavorano per una redazione di una testata giornalistica. Per semplicità gli utenti prelevati dal database lavorano per una redazione giornalistica qualsiasi, scelta staticamente.

Viene riportato di seguito lo schema ER del database.



L'importazione dei dati dal database relazionale è stata realizzata mediante 7 regole di mapping, ognuna delle quali si occupa di mappare una porzione dei dati. Questo metodo è stato adottato per semplificare la comprensione delle regole e facilitarne l'implementazione. Sono state inoltre create 3 query sparql per la visualizzazione dei videogiochi con le informazioni complete e delle recensioni rilasciate per essi, utilizzate per testare la correttezza delle regole di mapping e verificare eventuale inconsistenze.

Il reasoner utilizzato è OnTop versione 4.0.3.

Viene mostrata di seguito una panoramica delle regole di mapping e delle query Onto sparql:

```

mapping--553325724
:videogame_{id}a:Videogame::hasReleaseDate{releaseDate}::hasAverageRatingExpert{averageRatingExpert}::hasAverageRatingVideogamer
SELECT * FROM videogame

mapping--151113921
:videogame_{videogameID}a:Videogame::hasGenre:genre_{name}.
SELECT videogame.id as videogameID, videogame.title, genre.id as genreID, genre.name
FROM videogame join vgG ON videogame.id = vgG.videogame
join genre ON genre.id = vgG.genre

mapping--622383258
:genre_{genreName}a:Genre.
SELECT id, name as genreName FROM genre

mapping--1376823699
:videogame_{videogameID}a:Videogame::hasPlatform:piattaforma_{platformID}.
SELECT videogame.id as videogameID, videogame.title as vgTitle, platform.id as platformID, platform.name as platformName
FROM videogame join vgP ON videogame.id = vgP.videogame join platform ON platform.id = vgP.platform

mapping--1456650938
:worker_{workerID}a:Expert::hasUsername{username}.
SELECT videogame.id as videogameID, videogame.title, worker.id as workerID, worker.username, review.id as reviewID, review.score
FROM
review join worker ON review.worker = worker.id join videogame ON review.videogame = videogame.id

mapping--472902482
:piattaforma_{id}a:HomeConsole::platformName{platformName}.
SELECT id, name as platformName FROM platform

mapping--1213197377
:review_{reviewID}a:ExpertReview::reviewsAbout:videogame_{videogameID}::reviewMadeBy:worker_{workerID}::hasScore"9"^^xsd:integer.
SELECT videogame.id as videogameID, videogame.title, worker.id as workerID, worker.username, review.id as reviewID, review.score
FROM
review join worker ON review.worker = worker.id join videogame ON review.videogame = videogame.id

```

Come detto prima, le regole di mapping sono state scomposte in varie parti più semplici:

- mapping-622383258 per mappare i nuovi generi
- mapping--472902482 per mappare le nuove piattaforme ed assegnare loro il nome
- mapping-13776823699 assegna la piattaforma al videogioco
- mapping--151113921 assegna il genere al videogioco.
- mapping--553325724 registra il nuovo videogioco come appartenente alla classe Videogame e assegna ad esso tutte le sue caratteristiche nonché data property dell'ontologia (data di rilascio, voto medio della critica, voto medio dei videogiocatori).
- mapping-1456650938 registra il nuovo lavoratore con rispettivo username. Si noti che vengono selezionati dal database solo i lavoratori che hanno scritto delle recensioni.
- mapping-1213197377 assegna la recensione al videogioco con il rispettivo voto, specificando che essa è svolta dal lavoratore specifico.

Per semplicità si assume che tutti gli utenti siano lavoratori di qualche redazione e che quindi tutte le recensioni siano di tipo ExpertReview.

Si allegano come richiesto lo schema DDL per la definizione delle tabelle SQL, le regole di mapping (file regole-mapping.ttl) e triple materializzate (triple.ttl).



## 9. Applicazione Client

La realizzazione del client è avvenuta tramite il framework javascript Angular. L'app realizzata è una single web page application divisa per componenti (per semplicità un componente corrisponde ad una pagina dei mockup mostrati precedentemente). Vi sono dunque 4 components, chiamati home, videogame, review e platform, che mostrano rispettivamente il contenuto della home, la scheda del videogioco, la recensione nel dettaglio e la scheda della piattaforma. È presente anche un file chiamato `vgAwservice.ts` che ha il compito di effettuare le richieste via REST API alla LDP (è stato utilizzato GraphDB). In particolar modo, `vgAwservice` viene iniettato nel costruttore di ciascun componente e, in base ad esso, viene effettuata la corretta richiesta. La comunicazione avviene attraverso HTTP GET e il risultato prodotto dalla query viene restituito sottoforma di JSON.

Ad esempio, il component `videogame` chiederà al service di ottenere tutte le informazioni relative al videogioco selezionato. Il service, a sua volta, contatterà l'endpoint `sparql` e chiederà di eseguire la query opportuna e restituirà il risultato sottoforma di json. Infine, ottenuto il risultato, il component mostrerà il videogioco secondo lo stile proposto negli screenshots relativi all'esempio con dati reali.