

Cas d'ús de MongoDB

Bases de Dades no Relacionals

Grau en Matemàtica Computacional i Analítica de Dades

Víctor Benito Segura (1597165)

Mireia Majó i Cornet (1597716)

Àlex Martín González (1605489)

Albert Roca Llevadot (1603375)



Universitat Autònoma de Barcelona

Índex

<u>Introducció</u>	<u>2</u>
<u>Treball en equip</u>	<u>2</u>
<u>Treball previ</u>	<u>3</u>
<u>Requisits</u>	<u>3</u>
<u>Creació del repositori de Github</u>	<u>3</u>
<u>Patrons de disseny</u>	<u>4</u>
<u>Script de Python</u>	<u>7</u>
<u>Consultes</u>	<u>8</u>

Introducció

En aquest projecte de MongoDB, construirem una base de dades per a la botiga de còmics "Còmic Feliç". L'objectiu principal és crear una base de dades eficient que pugui ser utilitzada per l'empresa. Per aconseguir-ho, seguirem diversos passos per optimitzar el disseny i l'estructura de la base de dades:

En primer lloc, farem una anàlisi del model Entitat-Relació proposat on examinarem les entitats, els atributs i les relacions que s'estableixen.

Seguidament, aplicarem patrons de disseny per transformar el model Entitat-Relació en un conjunt de col·leccions que puguin ser emprades en un programari no relacional com NoSQLBooster. Per fer aquests pas, considerarem les consultes que s'han de realitzar a la botiga, les quals estan detallades al "Joc de Proves" i ens ajudaran a modelar la nostra base de dades.

Per a processar les dades del model relacional, farem servir un script de Python que ens permetrà transformar les dades originals en format CSV i convertir-les en col·leccions de MongoDB.

Finalment, executarem les consultes del Joc de Proves a través de NoSQLBooster per comprovar que la nostra base de dades és vàlida i compleix les necessitats de la botiga.

Treball en equip

El projecte l'hem treballat tots conjuntament, procurant que cadascú assumís el mateix volum de feina. En primer lloc, vam fer una reunió presencial on es va realitzar tot el treball previ a la programació del codi i de les consultes, on tots vam aportar idees per veure com tractàvem els requisits i el diagrama d'Entitat-Relació.

En segon lloc, vam dur a terme diverses videotrucades on vam fer el codi de Python tots junts, i finalment, un dels components actualitzava el programa al GitHub. Tot seguit, un cop el codi va estar finalitzat, ens vam repartir equitativament les consultes de la següent manera:

- Mireia: 1,2
- Albert: 3,8
- Àlex: 4,5,7
- Víctor: 6,9,10

En les ocasions on un dels components del grup tenia una consulta més senzilla, ajudava a comentar el codi. Per últim, ens hem repartit les seccions de l'informe final:

- Mireia: Patrons de Disseny, Script de Python i estructura del document
- Àlex: Treball en equip i Consultes
- Albert: Comentari codi main.py
- Víctor: Introducció, Treball Previ i README

Treball previ

Abans de començar a aplicar els patrons de disseny farem un estudi previ on visualitzarem la documentació del projecte.

Per tal d'entendre millor les necessitats de la botiga de còmics, llegirem els requisits que es van demanar i, a la següent secció, mostrarem el model Entitat-Relació facilitat per l'enunciat del projecte.

Requisits

Es vol fer una base de dades que ens permeti emmagatzemar les publicacions de col·leccions de llibres de diferents editorials que disposa una tenda de còmics.

Una editorial és una empresa que s'identifica pel nom i disposem del responsable, adreça i país i que crea col·leccions de publicacions. De fet, una mateixa col·lecció es pot crear en més d'una editorial.

De les col·leccions en sabem el nom, total d'exemplars, gènere o gèneres al que pertany (per exemple: terror, fantasia, etc) i l'editorial. A més, de l'idioma en el que s'ha redactat, l'any d'inici, l'any de finalització (si es que ha finalitzat) i un atribut que indica si la col·lecció ha acabat o no.

Cada col·lecció esta formada per diferents publicacions (llibres) que s'identifiquen amb l'ISBN. També cal guardar el títol, autor, número de pàgines, stock i preu a tenda. A cada llibre hi apareixen diversos personatges dels que volem guardar el nom i tipus. A més, aquests personatges poden aparèixer en més d'una publicació.

Per últim, guardarem els artistes que han participat en la creació de les publicacions. Dels artistes guardem el nom artístic, nom, cognoms, data de naixement i país. Aquests artistes poden participar tant com a guionistes com a dibuixants.

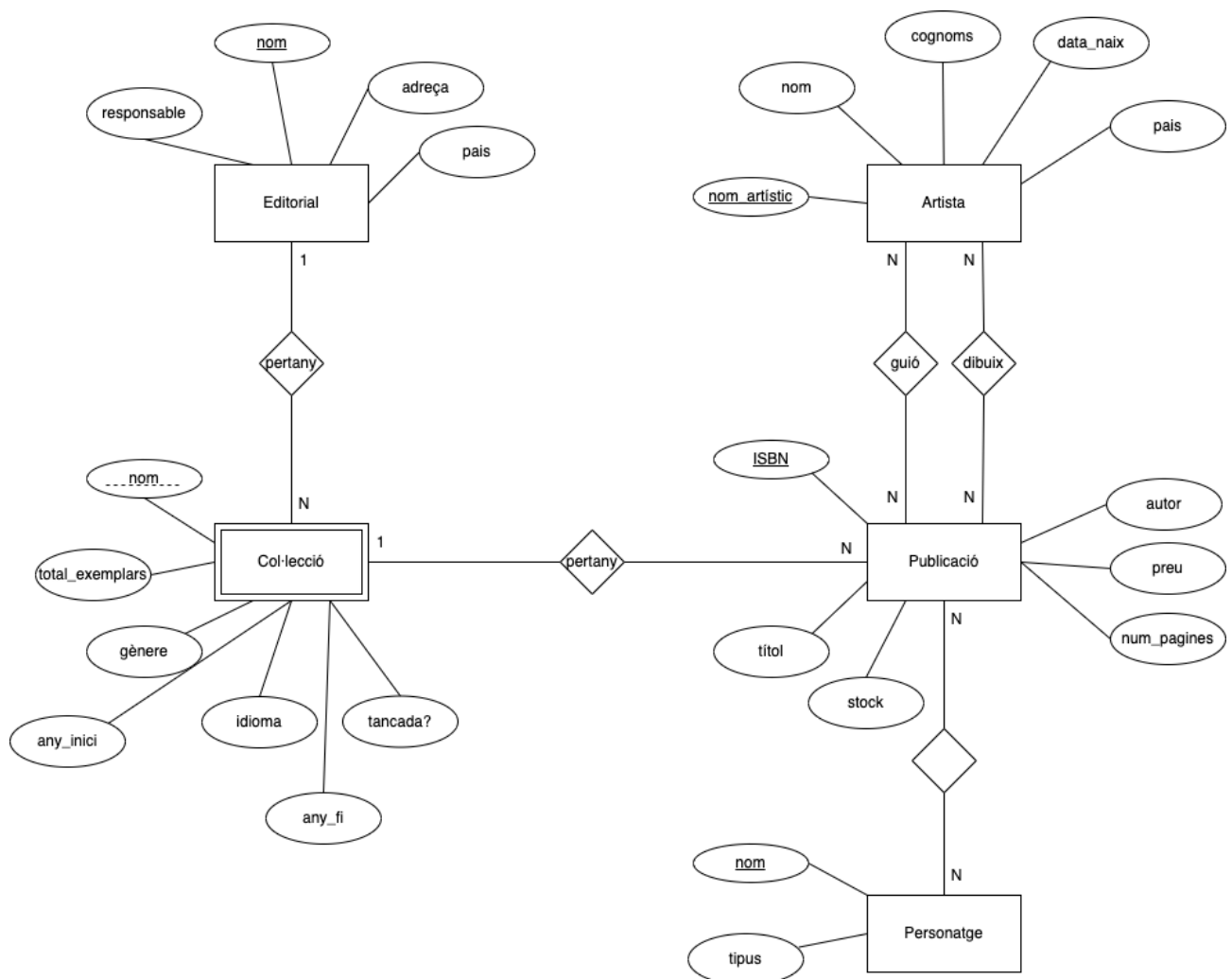
Creació del repositori de GitHub

Per concloure aquest apartat de treball previ, crearem un repositori a GitHub per tal d'anar actualitzant l'estat del projecte. En aquest repositori, realitzarem els commits de les diferents tasques que se'ns proposen: script de Python (`main.py`), les consultes realitzades al NoSQLBooster (`Consultes_Mongo DB.docx`) i aquest informe (`Informe Projecte MongoDB.docx`).

[Enllaç per accedir al repositori de GitHub](#)

Patrons de disseny

El diagrama d'entitat-relació que se'ns facilita per al nostre projecte és el següent. Fixem-nos que el diagrama consta de cinc entitats diferents:



En la base de dades no relacional que nosaltres hem plantejat, hem fet servir un total de 4 col·leccions. A continuació es mostren els noms de les quatre col·leccions i un exemple de document per a cadascuna d'elles.

Podeu observar que alguns camps contenen referències de documents de la resta de col·leccions. Els camps en qüestió (tant els que referencien com els que són referenciats) estan destacats amb colors per tal d'identificar-los clarament:

ARTISTES
<pre>{ "id" : ObjectId("643c5bb999304a89a96000d9"), "Nom_artistic" : "Artista1", "nom" : "nom1", "cognoms" : "cognoms1", "data_naix" : ISODate("1985-02-05T01:00:00.000+01:00"), "pais" : "Espanya" }</pre>

EDITORIALS
<pre>{ "_id" : ObjectId("643c5bb999304a89a96000d6"), "NomEditorial" : "Juniper Books", "resposable" : "Juniper Jonhs", "adreca" : "1501 Lee Hill Dr 1, Boulder, CO 80304", "pais" : "EEUU" }</pre>

COL·LECCIONS
<pre>{ "_id" : ObjectId("643c5bb999304a89a96000e0"), "NomColleccio" : "Lord of the Rings", "NomEditorial" : "Juniper Books", "genere" : ["fantasia", "belica"], "idioma" : "EN", "any_inici" : 1954, "any_fi" : 1955, "tancada" : true }</pre>

PUBLICACIONS
<pre>{ "_id" : ObjectId("643c5bba99304a89a96000fa"), "ISBN" : 23, "titol" : "Dracula", "stock" : 3, "autor" : "Bram Soker", "preu" : 125.5, "num_pagines" : 258, "guionistes" : ["Artista4"], "dibuixants" : ["Artista6", "Artista7"], "NomColleccio" : "Clasicos", "NomEditorial" : "Penguin", "personatges" : [{ "nom" : "Dracula", "tipus" : "protagonista", }] }</pre>

A continuació es detallen les estratègies utilitzades per tal d'establir relacions entre les col·leccions anteriors, totes elles amb les seves justificacions pertinents:

- **Relació editorials - col·leccions**

Si revisem les dades proporcionades en el fitxer `Dades.xlsx`, ens adonem hi ha editorials diferents que han publicat col·leccions amb el mateix nom. Per tant, el nom no és suficient per distingir una col·lecció d'una altra, sinó que ens cal també saber a quina editorial pertany (considerem que una mateixa editorial no pot publicar dues col·leccions amb el mateix nom). Per aquest motiu, les col·leccions tenen un camp anomenat `NomEditorial` que ens permetrà identificar a quina editorial pertanyen.

Com que es tracta de d'una relació 1-N (cada editorial té diverses col·leccions, però cada col·lecció tan sols pot pertànyer a una editorial), hem decidit referenciar el nom de l'editorial en un dels camps dels documents de la col·lecció "col·leccions".

- **Relacions col·leccions - publicacions i editorials - publicacions**

Al diagrama també hi observem que cada publicació pertany a una col·lecció (i cada col·lecció, al seu torn, pot tenir diverses publicacions). Per a aquesta relació (que és 1-N), hem decidit que cada publicació tindrà un camp amb el nom de la col·lecció a la qual pertany.

Per altra banda, les publicacions també contindran el camp `NomEditorial`. D'aquesta manera, ens serà possible identificar exactament a quina col·lecció pertany cadascuna de les publicacions, independentment de si el nom d'aquesta no és únic (ja que el nom de l'editorial ens permetrà distingir les col·leccions amb noms iguals).

Per tant, en aquests dos casos hem fet també ús de referències.

- **Relació publicacions - artistes**

Veiem que cada artista pot estar vinculat a una publicació a través de dos tipus de relació diferents: per una banda, de guió, i per l'altra, de dibuix. Per a representar-les, hem decidit també utilitzar referències dins dels documents de la publicació indicant els artistes que hi han treballat. Com que hi poden haver diversos guionistes i diversos dibuixants, els camps "guionistes" i "dibuixants" són llistes amb els noms dels artistes que hi han participat.

- **Relació publicacions - personatges**

Fixem-nos que l'entitat "personatge" tan sols té dos atributs: el nom i el tipus. A més, el tipus d'un mateix personatge pot canviar en funció de la publicació on apareix, ja que en diferents llibres pot prendre rols diferents.

Per aquests motius, hem decidit que la millor opció era encastar els personatges dins de la publicació. Així doncs, els documents de dins de la col·lecció "publicacions" consten d'un camp anomenat "personatges" que conté una llista de subdocuments. Cadascun d'aquests subdocuments té dos camps: en nom del personatge i el seu tipus.

L'ús de referències a l'hora de representar les relacions ens permet evitar problemes de duplicitat de dades i fa més senzilla l'actualització de documents. Per exemple, si una editorial canvia de seu i n'hem d'actualitzar l'adreça, tan sols ens caldrà editar aquesta informació del document corresponent dins de la col·lecció "editorial". Totes les publicacions que pertanyin a aquesta editorial no s'hauran d'editar perquè ja estaran referenciant el document actualitzat.

Script de Python

Aquest projecte consta també d'un programa en Python que s'encarrega de llegir les dades del fitxer `Dades.xlsx` i les puja a MongoDB. A continuació s'expliquen els passos que segueix el programa. Els números especificats abans de cada pas fan referència a les línies de codi on es troben:

- **[1 - 2] Imports necessaris**

Tan sols necessitem dues llibreries. Per una banda, Pandas ens resultarà útil per tal de llegir les dades del full de càlcul i transformar-les en un dataframe. Per altra banda, farem servir PyMongo per tal de pujar les dades a MongoDB.

- **[4 - 7] Lectura de les dades del full de càlcul**

Utilitzem la funció `read_excel()` de Pandas per tal de llegir les tres pàgines del document `Dades.xlsx`. Guardem les dades de cadascuna de les pàgines en dataframes diferents.

- **[10 - 21] Emmagatzematge de les dades en cadascuna de les col·leccions**

Indexem les columnes dels dataframes que contenen els camps que necessitem en cadascuna de les col·leccions. El format que farem servir és de diccionari.

- **[23 - 32] Addició dels personatges a publicacions**

Com que volem encastar les dades dels personatges dins de les publicacions, fem servir bucles que ens permeten emmagatzemar la informació de cada personatge dins del seu llibre corresponent. Coneixem a quines publicacions apareix cada personatge a través de l'ISBN.

- **[34 - 39] Format de les llistes**

Ens ha calgut també crear un bucle per tal que el MongoDB detectés les llistes amb el format correcte. Sense aquest fragment de codi, el MongoDB detectaria cada llista com un sol string.

- **[41 - 46] Eliminació del camp `any_fi`**

Com que les col·leccions que encara no estan tancades no tenen data de finalització, farem que aquest camp sigui inexistent en les col·leccions no acabades. Per fer-ho, utilitzem un bucle que assigna (o no) el camp `any_fi` a cada document en funció de si la col·lecció està tancada.

- **[48 - 60] Connexió amb la base de dades**

Establim les variables necessàries per la connexió i seleccionem la base de dades que farem servir (l'hem anomenat `llibreria`). Eliminem la versió anterior de la base de dades amb aquest mateix nom per tal de sobreescriure completament la informació.

- **[62 - 68] Inserció de les dades**

Per a cadascuna de les col·leccions, inserim les dades que hem estat netejant al llarg del programa. Així queden escrites al MongoDB i podrem hi podrem fer consultes a través del NoSQLBooster. Quan s'acaben de pujar els documents, el programa de Python tanca la connexió amb el MongoDB.

Consultes

1. Les 5 publicacions amb major preu. Mostrar només el títol i preu.

```
db.publicacions.aggregate()  
.project({_id:0, titol:1, preu: 1})  
.sort({preu:-1})  
.limit(5)
```

	titol	preu
1	Dracula	125.5
2	Tragedias	85.4
3	Romances	72.4
4	Crimen y castigo	59.4
5	En el Este	43.5

Podem observar com les cinc publicacions amb major preu són: Dracula, Tragedias, Romances, Crimen y castigo i En el Este amb un cost de 125.50€, 85.40€, 72.40€, 59.40€ i 43.50€ respectivament.

2. Valor màxim, mínim i mitjà del preus de les publicacions de l'editorial Juniper Books.

```
db.publicacions.aggregate()  
.match({NomEditorial: 'Juniper Books'})  
.group({  
  _id: null,  
  preu_minim: { $min: "$preu" },  
  preu_maxim: { $max: "$preu" },  
  preu_mitja: { $avg: "$preu" }})  
.project({_id:0})
```

	preu_maxim	preu_minim	preu_mitja
1	32.5	27.85	29.1182

Observem com el preu màxim de les publicacions de Juniper Books és 32.5 €. Pel que fa al preu mínim veiem que és 27.85 €. Finalment, el preu mitjà de totes les publicacions que té aquesta editorial és 29.12 €.

3. Artistes (nom artístic) que participen en més de 5 publicacions com a dibuixant.

```
db.publicacions.aggregate()  
.unwind("$dibuixants")  
.group({_id:"$dibuixants",count:{$sum:1}})  
.match({count:{$gt:5}})  
.project({_id:0,dibuixants:"$_id"})
```

	dibuixants
1	Artista1
2	Artista2

Els noms artístics dels artistes que participen en més de cinc publicacions com a dibuixants són l'Artista1 i l'Artista2.

4. Numero de col·leccions per gènere. Mostra gènere i número total.

```
db.collection.aggregate([  
  {$unwind: "$genere"},  
  {$group: {  
    _id: "$genere",  
    num_colleccions: {$sum: 1}  
  }},  
  {$project: { genere: "$_id", num_colleccions: 1, _id: 0 }}  
)
```

	colleccions	0.118 s	5 Docs
	genere	num_colleccions	
1	fantasia	4	
2	magia	2	
3	belica	2	
4	clasicos	1	
5	suspense	1	

Observem com el gènere amb més col·leccions és 'fantasia' amb quatre col·leccions, seguit de 'magia' i 'belica' amb dues i de 'clasicos' i 'suspense' amb una.

5. Per cada editorial, mostrar el recompte de col·leccions finalitzades i no finalitzades.

```
db.colleccions.aggregate([
  {
    $group: {
      _id: "$NomEditorial",
      ColleccioTancada: { $sum: { $cond: [ "$tancada", 1, 0 ] } },
      ColleccioOberta: { $sum: { $cond: [ { $not: "$tancada" }, 1, 0 ] } },
    }
  }
])
```

	_id	ColleccioTancada	ColleccioOberta
1	Juniper Books	2	0
2	Penguin	1	1
3	The Folio Society	1	0

Per una banda, veiem com l'editorial Juniper Books té dues col·leccions tancades i no en té cap d'oberta. per l'altra banda, l'editorial Penguin té una col·lecció oberta i una tancada. Finalment l'editorial The Folio Society només té una col·lecció tancada.

6. Mostrar les 2 col·leccions ja finalitzades amb més publicacions. Mostrar editorial i nom col·lecció.


```
db.colleccions.aggregate([
  {
    $lookup: {
      from: "publicacions",
      localField: "NomColleccio",
      foreignField: "NomColleccio",
      as: "publicacions"
    }
  },
  { $match: { tancada: true } },
  { $unwind: "$publicacions" },
  { $group: {
    _id: { NomColleccio: "$NomColleccio", NomEditorial:
"$publicacions.NomEditorial" },
    NumPublicacions: { $sum: 1 }
  } },
  { $sort: { NumPublicacions: -1 } },
  { $project: {
    _id: 0,
    NomColleccio: "$_id.NomColleccio",
    NomEditorial: "$_id.NomEditorial",
  } },
  { $limit: 2 }
])
```

	colleccions	0.033 s	2 Docs
	NomColleccio ▾	NomEditorial ▴	
1	Harry Potter	Juniper Books	
2	Harry Potter	Penguin	

Veiem que les dues col·leccions que ja han finalitzat i que tenen més publicacions són de Harry Potter. En aquest cas, aquesta col·lecció la porten dues editorials, per tant, veiem que ens apareix la col·lecció i l'editorial Penguin i Juniper Books com a solució.

7. Mostrar el país d'origen de l'artista o artistes que han fet més guions.

```
db.publicacions.aggregate([
  { $unwind: "$guionistes"},
  { $group:
    {
      _id: "$guionistes",
      count: { $sum: 1}
    }
  },
  { $sort: { count: -1 } },
  { $limit: 1},
  {
    $lookup: {
      from: "artistes",
      localField: "_id",
      foreignField: "Nom_artistic",
      as: "artistes"
    }
  },
  { $project: { _id: 0, pais: {$first: "$artistes.pais" }} }
])
```

	publicacions		0.049 s	1 Doc
	pais			
1	Noruega			

Veiem que l'artista que més guions ha fet és d'origen noruec.

8. Mostrar les publicacions amb tots els personatges de tipus “heroe”.

```
db.publicacions.aggregate()  
  .match({"personatges.tipus":"heroe"})  
  .project({_id:0,ISBN:1,  
    nHeroes:{$size:{$filter:{input:"$personatges",  
cond:{$eq:["$$this.tipus","heroe"]}}}},  
    nPersonatges:{$size:"$personatges"}  
  })  
  .match({$expr:{$eq:["$nHeroes","$nPersonatges"]}})  
  .project({_id:0,ISBN:1})
```

	ISBN
1	4
2	20
3	22

Observem que les publicacions amb tots els personatges de tipus ‘heroe’ són les que corresponen als ISBNs 4, 20 i 22 de la nostra base de dades.

9. Modificar el preu de les publicacions amb stock superior a 20 exemplars i incrementar-lo un 25%.

Abans de l'update:

```
db.publicacions.find().projection({'_id':0,'stock':1,'preu':1}).limit(7)
```

	stock	preu
1	20	32,5.0
2	5	32,5.0
3	50	32,5.0
4	7	27,85.0
5	6	27,85.0
6	2	27,85.0
7	22	27,85.0

Després de l'update:

```
db.publicacions.updateMany(  
  { "stock": { $gt: 20 } },  
  { $mul: { "preu": 1.25 } }  
)
```

Key	Value	Type
(1)	{ acknowledged : true, matchedCount : 7, modifiedCount : 7 }	Object
acknowledged	true	Bool
matchedCount	7	Int32
modifiedCount	7	Int32

	stock	preu
1	20	32,5.0
2	5	32,5.0
3	50	40,625.0
4	7	27,85.0
5	6	27,85.0
6	2	27,85.0
7	22	34,8125.0

Observem com s'han modificat set publicacions les quals tenien un estoc superior a vint unitats i se'ls ha augmentat el preu un 25%.

10. Mostrar ISBN i títol de les publicacions conjuntament amb tota la seva informació dels personatges.

```
db.publicacions.find().projection({'_id':0, 'ISBN':1, 'titol':1, 'personatges':1})
```

Key	Value	Type
▲ (1)	{ ISBN : 1, titol : "The fellowship of the ring" } (3 fields)	Object
ISBN	1	Int32
titol	The fellowship of the ring	String
▲ personatges	Array[3]	Array
▶ 0	{ nom : "Gandalf", tipus : "mago" }	Object
▶ 1	{ nom : "Frodo", tipus : "heroe" }	Object
▶ 2	{ nom : "Samsagaz", tipus : "segundo" }	Object
▶ (2)	{ ISBN : 2, titol : "The two towers" } (3 fields)	Object
▶ (3)	{ ISBN : 3, titol : "The return of the King" } (3 fields)	Object
▶ (4)	{ ISBN : 4, titol : "Harry potter y la piedra filosofal" } (3 fields)	Object
▶ (5)	{ ISBN : 5, titol : "Harry potter y la camara secreta" } (3 fields)	Object
▶ (6)	{ ISBN : 6, titol : "Harry potter y el prisionero de Azhaban" } (3 fields)	Object
▶ (7)	{ ISBN : 7, titol : "Harry potter y el caliz de fuego" } (3 fields)	Object

Per cada publicació podem veure l'ISBN el títol, els personatges que hi apareixen i de quin tipus són.