**INFORME PROJECTE MONGODB**

**ÍNDEX**

Introducció

Treball en equip

Treball previ

* Requisits
* Model Entitat-Relació
* Creació repositori Github

Patrons de disseny

Script de Python

Consultes

**Introducció**

En aquest projecte de MongoDB construirem una base de dades per la botiga de còmics "Còmic Feliç". Per tal d'aconseguir una base de dades eficient que pugui ser utilitzada per l'empresa, realitzarem diversos passos:

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

Seguidament, aplicarem patrons de disseny per transformar el model Entitat-Relació en un conjunt de col·leccions que puguin ser utilitzades en un programari no relacional com NoSQLBooster. Per fer aquests pas, considerarem les consultes que s'han de realitzar a la botiga. Aquestes consultes les trobarem al "Joc de Proves" i ens ajudaran a modelar la nostra base de dades. Mitjançant un script de Python, rebrem les dades del model relacional en format CSV i enviarem les dades tractades en col·leccions al servidor de MongoDB.

Finalment, executarem les consultes del Joc de Proves a NoSQLBooster per comprovar que la base de dades pot ser usada correctament a la nostra botiga.

**Treball en equip**

TODO (Àlex) Explicar distribució de la feina

* + Videotrucada per fer el codi de python junts
  + Distribució equitativa de les consultes

**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 mostrarem el model Entitat-Relació que es va construir prèviament.

1. 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.

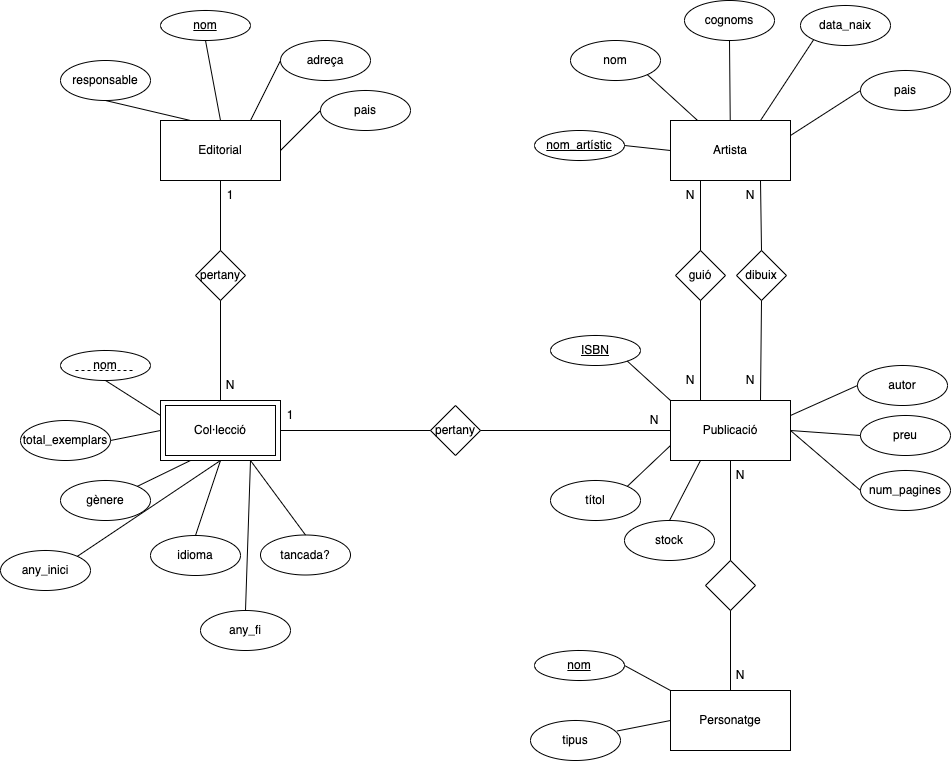
Una editorial és una empresa que s’identifica pel nom i disposem del responsable, adreça i país i que crea col·leccions de publicacions. De fet, una mateixa col·lecció es pot crear en més d’una editorial.

De les col·leccions en sabem el nom, total d’exemplars, gènere o gèneres al que pertany (per exemple: terror, fantasia, etc). 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.

1. Model Entitat-Relació



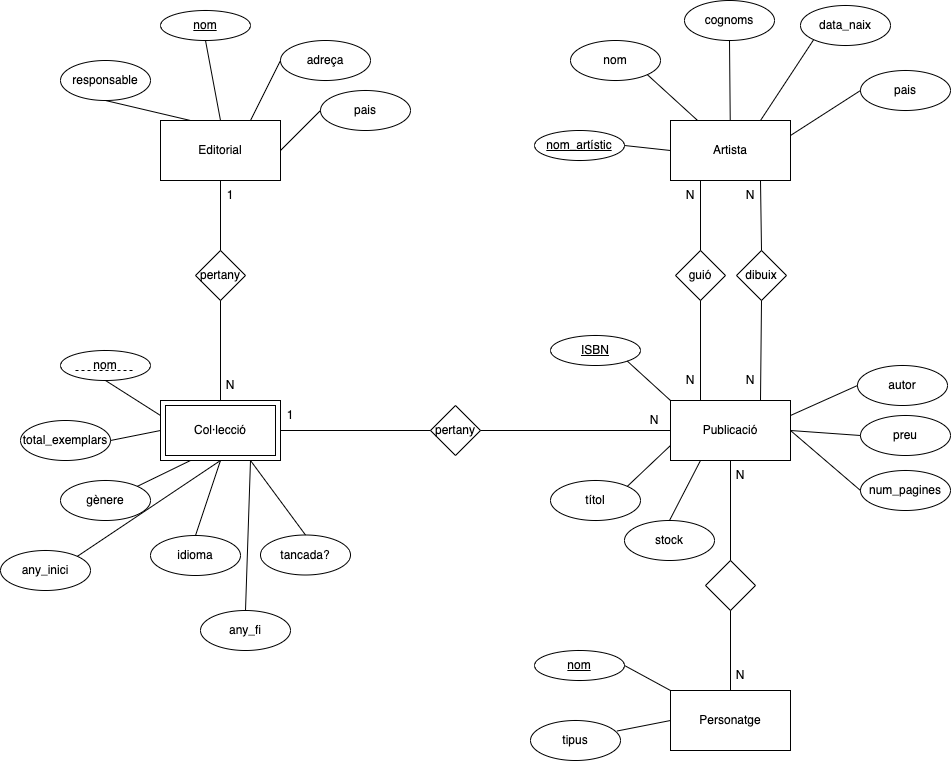
1. Creació repositori github

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), consultes (Consultes\_Mongo DB.docx) i informe(Informe Projecte Mongo.docx).

[Link](https://github.com/AlbertRoca29/MongoDB) per accedir al repositori 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 els documents de la col·lecció “publicacions” contenen referències de documents de la resta de col·leccions. Els camps en qüestió estan destacats amb colors per tal d’identificar-los clarament:

| ARTISTES |
| --- |
| {  **"\_id"** : ObjectId("643c5bb999304a89a96000d9"),  **"Nom\_artistic"** : "Artista1",  **"nom"** : "nom1",  **"cognoms"** : "cognoms1",  **"data\_naix"** : ISODate("1985-02-05T01:00:00.000+01:00"),  **"pais"** : "Espanya"  } |

| EDITORIALS |
| --- |
| {  **"\_id"** : ObjectId("643c5bb999304a89a96000d6"),  **"NomEditorial"** : "Juniper Books",  **"resposable"** : "Juniper Jonhs",  **"adreca"** : "1501 Lee Hill Dr 1, Boulder, CO 80304",  **"pais"** : "EEUU"  } |

| COL·LECCIONS |
| --- |
| {  **"\_id"** : ObjectId("643c5bb999304a89a96000e0"),  **"NomColleccio"** : "Lord of the Rings",  **"genere"** : [ "fantasia", "belica" ],  **"idioma"** : "EN",  **"any\_inici"** : 1954,  **"any\_fi"** : 1955,  **"tancada"** : true  } |

| PUBLICACIONS |
| --- |
| {  **"\_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",  }  ]  } |

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 - publicacions:

Observant el diagrama, ens adonem que les entitats “editorial” i “publicació” ni tan sols estan relacionades directament, sinó que es comuniquen a través de l’entitat “col·lecció”. No obstant, si revisem les dades proporcionades en el fitxer Dades.xlsx, ens adonem que dins d’una mateixa col·lecció hi podem trobar llibres publicats per diferents editorials. Per aquest motiu, hem decidit que la col·lecció “col·leccions” no inclourà les editorials a les quals pertany, sinó que aquesta informació es trobarà directament dins les publicacions.

Com que es tracta d’una relació 1-N (cada editorial té diverses publicacions, però cada publicació pertany a una única editorial), hem decidit referenciar l’editorial (a través del nom) dins de cada document de la col·lecció “publicacions”.

* Relació col·leccions - 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 tant, hem utilitzat un altre cop una referència.

Si ajuntem aquest punt amb l’anterior, el resultat és que cada document de la col·lecció “publicacions” tindrà un camp amb el nom de l’editorial a la qual pertany i un camp amb el nom de la col·lecció a la qual pertany, referenciant així els documents corresponents de les col·leccions “editorials” i “col·leccions”, respectivament.

* Relació publicacions - artistes

Veiem que cada artista pot estar relacionat amb 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 “guonistes” 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 es 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**

TODO (Àlex) Consultes amb outputs i explicacions