

# Visualização de Dados de Informações Extraídas da Web - Um Estudo sobre Conteúdo Popular Japonês

Gabriel Fontenelle Senno Silva<sup>1</sup>

<sup>1</sup>Centro Universitário Senac - Campus Santo Amaro (SENAC-SP)  
Av. Engenheiro Eusébio Stevaux, 823 – São Paulo – CEP 04696-000 – SP – Brasil

colecionador.gabriel@gmail.com

**Resumo.** Primeiramente falamos sobre alguns dos conteúdos pertencentes a cultura popular japonesa e apresentamos os websites utilizados para extração de conteúdo, depois apresentamos em detalhes a modelagem do banco de dados, discutimos a implementação do crawler e o desenvolvimento de visualizações de dados e por fim mostramos as visualizações produzidas.

## 1. Introdução

Visualização de dados é uma forma de comunicar visualmente informações, com base em um conjunto de dados, que de outra forma não seriam facilmente identificáveis.

O foco deste trabalho é a criação de visualizações a partir de dados extraídos de websites que disponibilizam quantidade maciça de conteúdo. Neste trabalho dados foram obtidos automaticamente de diversos websites com o uso de uma biblioteca de crawling, que navega nos websites, e salvos em um banco de dados previamente modelado para geração posterior de visualizações.

Para a criação da visualização de dados foi escolhido o tema: cultura popular japonesa. A cultura popular japonesa é conhecida pelo desenvolvimento de animações, revistas em quadrinhos e gêneros literários influenciados por um estilo de desenho único focado nas expressões de suas personagens. O tema é abrangente e poderíamos obter e utilizar dados sobre os seus diversos produtos comercializados: mangás, animes, Light Novels, Visual Novels e outros bens de consumo derivados como braceletes e figuras de ação que são adquiridos por colecionadores em grande parte do mundo ocidental.

Os seguintes itens dessa cultura popular serviram como base para a modelagem do banco de dados e ajudaram na escolha dos websites para o crawler:

**Anime** significa animação em japonês, porém no ocidente é usado para se referir às animações provenientes do Japão. O termo pode ser usado para diversas animações como séries, filmes e OVA (*Original Video Animation* - animação distribuída direto em DVD ou BluRay).

**Mangá** significa história em quadrinhos em japonês, porém no ocidente é usado para referenciar histórias em quadrinhos provenientes do Japão ou que possuem o mesmo estilo estético.

**Light Novel** é um gênero literário caracterizado pelo menor número de páginas e escrita mais clara. Possui histórias fluídas com desenvolvimento rápido, muitas vezes utilizando-se de efeitos sonoros para ilustrar situações em vez de uma descrição completa da situação, como exemplo a saída de uma personagem de um quarto com uma batida grosseira da porta pode ser demonstrada apenas mencionando que a personagem saiu e o efeito sonoro da porta.

**Visual Novel** assim como Light Novel é um gênero literário originado no Japão, que é formado por ilustrações e textos mais claros de serem entendidos que livros tradicionais. Mas, diferente de Light Novels, se assemelham mais a um jogo de computador em que decisões dos jogadores podem alterar o rumo da história.

**Figuras de ação** são esculturas de personagens de animes, de jogos eletrônicos ou até mesmo Light Novels, produzidas geralmente em PVC, podendo ser ou não pintadas à mão. Ocionalmente são produzidas em quantidade limitada, e algumas edições são inclusive enumeradas tendo um valor maior para colecionadores.

Muito desses itens fazem parte de franquias, ou coleções, que é um conjunto que engloba histórias e personagens existentes em um mesmo universo fictício, podendo ser composta por livros com histórias seqüenciais ou por livros com histórias que complementa a compreensão do universo fictício.

O universo fictício nem sempre é limitado pela produção de conteúdo em uma única mídia como livros, uma continuação pode ser disponibilizada em jogos de computador, jogos eletrônicos, mangás, animes e até mesmo em história em áudio. Uma franquia conhecida por sua propagação em diversas mídias é a *.Hack*, que teve inicio com o anime *.hack//Sign*, mas continuações da história foram lançadas em jogos para Playstation e mangás.

Muitas franquias possuem venda de produtos como cartazes promocionais, adesivos, chaveiros, braceletes e outros acessórios baseados em seus personagens.

## 2. Desenvolvimento

### 2.1. Websites para extração de dados

Existem um grande número de websites que poderíamos utilizar como fonte para a extração de dados. Escolhemos os websites mais populares e que possuem grande quantidade de dados, para que pudessemos a partir dos dados extraídos gerar informações sobre franquias, como exemplo a quantidade de itens presentes em cada franquia. Alguns desses websites são similares e possuem conteúdo redundante. A escolha de websites com informações redundantes foi proposital uma vez que não poderíamos garantir a disponibilidade com os sites durante o desenvolvimento deste trabalho, de fato entre os websites escolhidos, que podem ser conferidos na lista abaixo, durante o desenvolvimento do sistema de crawling o website Manga-Updates ficou fora do ar durante alguns dias logo após o término do algoritmo para extração de seu conteúdo. Não só o website Manga-Updates como também o website AnimeBlade ficou indisponível, porém esse último teve sua hospedagem cancelada e não retornou.

<http://mangaupdates.com/> possui informações de mangás e Light Novels. Outras informações relacionadas como editoras, ilustradores e autores também foram extraídas do website.

<http://myfigurecollection.net/> possui informações de mangás, figuras de ação e outros produtos baseados em personagens.

<http://www.animecharactersdatabase.com> possui informações de jogos de computador, animes e personagens. Também foi possível extrair informações sobre dubladores de personagens de animes e de jogos de computadores.

<http://old.animeblade.com.br/> possuía informação de mangás e Light Novels com conteúdo em português.

Esses websites foram testados e apenas o <http://www.animecharactersdatabase.com/> demonstrou limitação quanto a quantidade de requisições, permitindo apenas 1600 requisições por hora.

Dentre os websites escolhidos, alguns possuem informações em outros idiomas além do idioma principal do website, como os títulos de mangás e animes. Para aproveitarmos essas informações o banco de dados foi modelado de forma a permitir múltiplos idiomas.

Decidimos extrair a maior quantidade possível de dados disponíveis nos websites escolhidos para que pudessemos criar diversas visualizações.

Como não podemos garantir o funcionamento dos websites escolhidos no momento de leitura desse documentos a seguir pode ser conferido *screenshots* das páginas utilizadas na extração de conteúdo.

The screenshot shows a web page from the Manga-Updates website. At the top, there's a header with the site's logo and navigation links. On the left, there's a sidebar with a poll, an app download link, and a small cartoon character. The main content area is titled "Publishers" and shows details for "Asahi Shimbunsha". It includes sections for "Alternate Names", "Notes", "Website", "Type", "Last Updated", "Publications", and "Series". A right sidebar contains links for various manga-related categories like News, Series Stats, and Members. The overall layout is clean with a light color scheme and some anime-style illustrations in the background.

**Figura 1.** Página do website Manga-Updates exibindo dados de uma editora.

Baka-Updates Manga

You are currently logged in as: Teste2352 [Logout]

**Manga Info**

**The Breaker - New Waves**

Add to reading list Add to wish list Add to: Select...

**Description [Edit]**  
In the aftermath of the desperate battle between Goomoonyong and the Martial Arts Alliance – Yi Shioon's ki-center was destroyed by his own master Goomoonyong, leaving him unable to practice martial arts...

**Original Webtoon**

**Type [Edit]**  
Manhwa

**Related Series [Edit]**  
[The Breaker \(Prequel\)](#)

**Associated Names [Edit]**  
Сокрушитель: Новые волны  
브레이커2  
브레이커NW  
The Breaker 2  
The Breaker: New Waves

**Groups Scanning**  
A-Team  
A3S Scans  
EPIC WORKS  
Ghasitly Scans  
More...

**Latest Release(s)**  
c. 1/1 by MangaCow (5d ago)  
c. 1/80 by MangaCow (12d ago)  
c. 1/79 by MangaCow (16d ago)  
[Search for all releases of this series](#)

**Status in Country of Origin [Edit]**  
12 Volumes (Ongoing)

**Completely Scanned? [Edit]**  
No

**Anime Start/End Chapter [Edit]**  
N/A

**User Reviews**  
N/A

**Forum**  
30 topics, 281 posts  
[Click here to view the forum](#)

**User Rating**  
Average: 8.8 / 10.0 (1605 votes)  
Bayesian Average: 8.76 / 10.0

10+	45% (729 votes)
9+	23% (376 votes)
8+	15% (245 votes)
7+	8% (121 votes)
6+	3% (44 votes)
5+	2% (28 votes)
4+	0% (8 votes)
3+	1% (13 votes)
2+	1% (14 votes)
1+	2% (27 votes)

**Last Updated**  
November 21st 2014, 9:35am GMT-3

**Sponsored Links**

**Image [Report inappropriate Content] [Edit]**

**Genre [Edit]**  
Action Comedy Drama Ecchi Martial Arts School Life Shounen  
[Search for series of same genre\(s\)](#)

**Categories [Edit]**  
[Vote these categories](#) Show all (some hidden)

- **Bodyguard/s Disciple**
- **Master-Disciple Relationship**
- **Romantic Subplot**
- **Secret Organization/s**
- **Special Ability/ies**
- **Special Technique**
- **Sudden Strength Gain**
- **Underworld Society**
- **Weak to Strong**

**Category Recommendations**  
[The Breaker](#)  
Arifureta Shokugyou de Sekai Saikyou (Novel)  
Seirei Tsukai no Kenbu (Novel)  
Suterareta Yussha no Eiyutan (Novel)  
Asobi ni Iku vol.1 (Novel)

**Recommendations**  
[The Breaker](#)  
Noblesse  
Mx0  
Superior Day  
Hagane no Renkinjutsushi  
More...

**Author(s) [Edit]**  
JEON Geuk-jin

**Artist(s) [Edit]**  
PARK Jin-Hwan

**Year [Edit]**  
2010

**Original Publisher [Edit]**  
Dawon (paperback volumes)  
Daum (digital)

**Serialized in (magazine) [Edit]**  
Daum (Daum)

**Licensed (in English) [Edit]**  
No

**English Publisher [Edit]**  
N/A

**Activity Stats (vs. other series)**  
Weekly Pos #201 ▼(-143)  
Monthly Pos #25 ▼(-9)  
3 Month Pos #22 ▼(-4)  
6 Month Pos #11 ▼(-5)

**List Stats**  
On 7558 reading lists  
On 742 wish lists  
On 92 unfinished lists  
On 412 custom lists

*Note: You must be logged in to update information on this page.*

**Manga Search**  
 Go

**MANGA.FU**  
News  
What's New!  
Series Stats  
Forums  
Chat

**Releases**  
Scantlators  
Series Info  
Mangaka  
Publishers  
Reviews

**Genres**  
Categories

**FAQ**  
Affiliates  
Members  
BU Forum  
Baka-Updates

**MEMBERS**  
User CP  
My Inbox  
My Lists

**TEAM-BU**  
Admin CP  
About Us

**Figura 2.** Página do website Manga-Updates exibindo dados de uma série. Neste caso uma história em quadrinhos proveniente da Coréia do Sul, referenciado como Manhwa no ocidente. Manhwa é o termo equivalente a história em quadrinhos no idioma coreano.

Baka-Updates

# Manga

You are currently logged in as: [Teste2352](#) [[Logout](#)]

**App**  
Try out our new iPhone application!

 Available on the App Store

**Manga Poll**  
Which kind of story would you find more interesting?

Dystopia  
 Utopia

[Vote](#) [Result](#) [See Old Polls](#)



Manga is the Japanese equivalent of comics with a unique style and following. Join the revolution! Read some manga today!

Coded in CONTEXT  
Join #baka-updates @irc.irchighway.net

[RSS Feed](#)

## Mangaka

### KAMIYA Yuu (榎宮祐) [\[Edit\]](#)

**Image** [\[Edit\]](#)  


**Comments** [\[Edit\]](#)  
 Married to [Hiragi Mashiro](#).

Thiago Furukawa Lucas, who is better known under his pseudonym, Yū Kamiya, was born in Brazil. He is the first foreign manga artist to make it big in Japan from Brazil.

**Blood Type** [\[Edit\]](#)  
 N/A

**Gender** [\[Edit\]](#)  
 Male

**Genres**  
 Fantasy(8) Shounen(6) Supernatural(6) Action(4) Ecchi(4) Sci-fi(4) Comedy(3) Doujinshi(3) Romance(3) Adventure(2) Seinen(2) Drama(1) Gender Bender(1) Harem(1) Hentai(1) Lolicon(1) Mature(1) Shoujo Ai(1) Yuri(1)

**Name (in native language)** [\[Edit\]](#)  
**榎宮祐**

**Birth Place** [\[Edit\]](#)  
 Brazil

**Birth Date** [\[Edit\]](#)  
 November 10, 1984

**Zodiac**  
 Scorpio

**Last Updated**  
 July 16th, 2:57pm GMT-3

Series Title (Click for series info)	Genre	Year
Clockwork Planet	Action, Fantasy, Sci-fi, Shounen	2013
Clockwork Planet (Novel)	Drama, Fantasy, Romance, Sci-fi, Shounen	2013
EArTh	Action, Ecchi, Sci-fi, Shounen, Supernatural	2006
EArTh ☾	Action, Ecchi, Sci-fi, Shounen, Supernatural	2010
Greed Packet ☾	Action, Comedy, Ecchi, Fantasy, Gender Bender, Lolicon, Mature, Seinen, Supernatural	2008
Itsuka Tenma no Kuro Usagi (Novel)	Fantasy, Romance, Shounen, Supernatural	2008
No Game No Life	Adventure, Comedy, Ecchi, Fantasy, Seinen, Supernatural	2013
No Game No Life (Novel)	Adventure, Comedy, Fantasy, Harem, Romance, Shounen, Supernatural	2012
Touhou dj - Evening Primrose - Free Heart	Doujinshi, Fantasy	2005
Touhou dj - Higashi no Kuni no Nemuranai	Doujinshi, Fantasy, Shoujo Ai	2006
UUtage	Doujinshi, Hentai, Yuri	2006

*Note: You must be logged in to update information on this page.*

**Manga Search**


**MANGA Fu**

[News](#)  
[What's New!](#)  
[Series Stats](#)  
[Forums](#)  
[Chat](#)  
[Releases](#)  
[Scanlators](#)  
[Series Info](#)  
[Mangaka](#)  
[Publishers](#)  
[Reviews](#)  
[Genres](#)  
[Categories](#)  
[FAQ](#)  
[Affiliates](#)  
[Members](#)  
[BU Forum](#)  
[Baka-Updates](#)

**MEMBERS**

[User CP](#)  
[My Inbox](#)  
[My Lists](#)

**TEAM-BU**

[Admin CP](#)  
[About Us](#)

Figura 3. Página do website Manga-Updates exibindo dados de um autor.

The screenshot displays a detailed character profile for the anime 'Cardfight!! Vanguard' on the AnimeCharacterDatabase. At the top, there's a navigation bar with links like 'Menu', 'Random Adams', 'Random Mals', 'Random Fennel', 'Random Game', 'Lounge 3', 'Who', 'New Quizzes', 'VS', 'Quids', 'Forum', and 'Search'. A banner for 'Claro fixo' with the tagline 'Fale muito e não caia duro com a conta de telefone.' is visible. On the right side, there's a sidebar for 'marisa.com.br' featuring various fashion items and their prices.

**Character Profile:**

- Title:** CARDFIGHT!! VANGUARD (SERIES)
- Series ID:** 102318
- English Title:** Cardfight!! Vanguard (Series)
- Original Title:** Kaada futsu!! Vangaido
- Finnegan Title:** カードファイト!! ヴァンガード
- Japanese Title:** カードファイト!! ヴァンガード
- Japanese Studio Name:** TMS Entertainment
- English Studio Name:** E10+ - Everyone ten and up
- Content Rating:** G
- Genre:** RPG
- Release Date:** 2001/01/08
- Links:** Home Page EN Wiki JP Wiki
- Character Popularity:** 8

**Story & Information:** This section contains a 'Note to Self' area where users can leave private notes, a 'Post' button to add a public note, and a 'View All Notes' link.

**Franchise Listing:** Includes links to related franchises: Cardfight!! Vanguard, Cardfight!! Vanguard: Asia Circuit Hen, Cardfight!! Vanguard: Link Joker Hen, and Cardfight Vanguard G.

**Additional Images:** There are 0 images available for upload. A placeholder message says 'Upload an Additional Image' and provides a file selection field.

**Related:** Shows four related anime titles: Cardfight!! Vanguard (Anime), Cardfight!! Vanguard (Anime), Cardfight!! Vanguard (Anime), and Cardfight Vanguard G (Anime).

**Forum Posts:** A sidebar on the right shows recent forum posts from the 'VS' section, including topics about 'Done Surgery', 'Injury Update', and 'Is this girl canon?'. It also includes a link to 'More Forum Posts'.

Figura 4. Página do website AnimeCharacterDatabase exibindo dados de um anime.

The screenshot displays the AnimeCharacterDatabase website interface. At the top, there's a navigation bar with various filters like 'Gender', 'Eye Color', 'Hair Color', 'Hair Length', 'Apparent Age', 'Animal Ears', and a search bar. A large banner for 'Recruit Teste2352' is visible. Below the banner, a promotional banner for 'AliOfertas Brasil' is shown.

The main content area features a profile for 'KARU YAMAJI'. It includes a thumbnail image of the character, a 'Note to Self' section with a text input field and a 'Post' button, and a 'View All Notes' link. Below this is a 'Cardfight Vanguard G' section with a character illustration holding a tablet.

The 'Profile' tab shows detailed information about Karu Yamaji, such as her ID (70423), name (Karu Yamaji), Japanese name (山崩川リカ), and voice actress (Kotomi Ootsuka). The 'Extra Info' tab lists her appearance details: eye color (Blue), hair color (Brown), hair length (To Neck), apparent age (Teen), gender (Female), and animal ears (No).

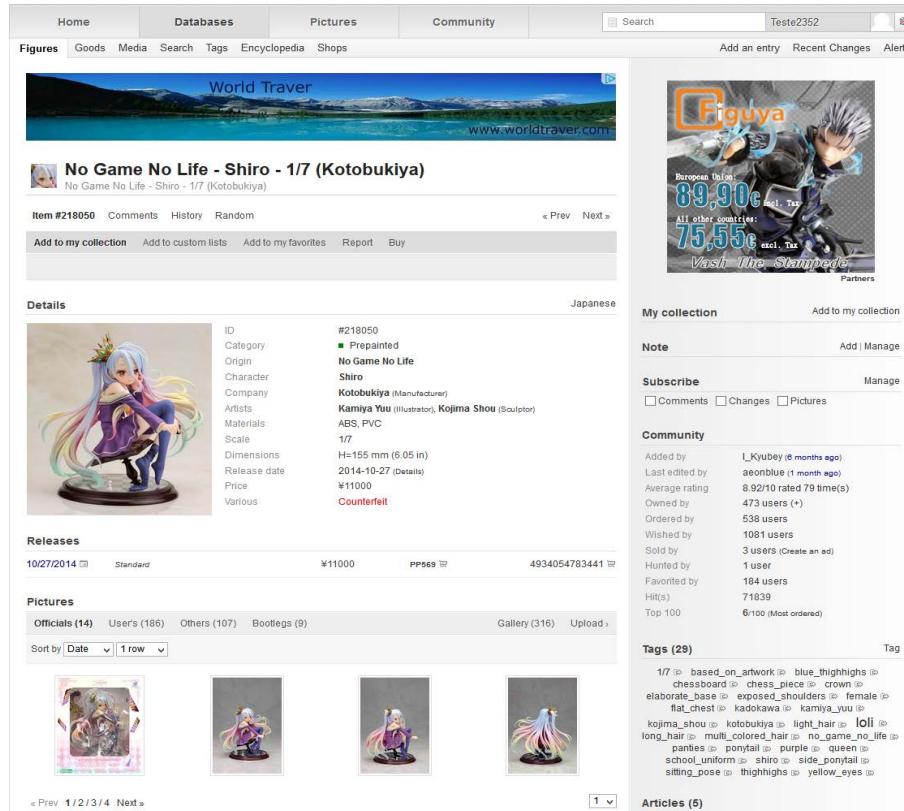
The 'KNOWN STATS' section provides a breakdown of character popularity: Unknown (0), Common (0), and Famous (0). There are also links for 'View Collected', 'My Cards', 'Discard', 'Trade', and 'Buy Cards'.

A 'FRIEND LIST' sidebar on the right shows friends like 'SERV TIME' (3:49 PM, Wed Nov 26 2014) and 'FAVORITES' (few others, few series).

**Figura 5.** Página do website AnimeCharacterDatabase exibindo dados de uma personagem.

The screenshot shows a product page from the website MyFigureCollection. At the top, there is a large image of a character from "No Game No Life" named Shiro, surrounded by butterflies. Below the header, there is a navigation bar with links for Home, Databases, Pictures, Community, Search, and Alerts. The main content area displays a product listing for a "No Game No Life - Shiro - Can Badge Strap". The listing includes a thumbnail image of the badge, its ID (#218380), category (Hanged up), classification (Can Badge Strap), origin (No Game No Life), character (Shiro), company (Contents Seed), materials (Plastic, Tin), dimensions (W=32 mm (1.26 in) L=110 mm (4.29 in)), release date (2014-06), and price (¥400). To the right of the product details, there is a sidebar with advertisements for MailChimp (scalable, reliable, secure) and Plamoya Japan (41000 items, Rare/Limited/Exclusive, up to 70% off). The sidebar also includes sections for My collection, Note, Subscribe, Community, Tags, and Related Clubs.

**Figura 6.** Página do website MyFigureCollection exibindo dados de um produto.



**Figura 7.** Página do website MyFigureCollection exibindo dados de uma figura de ação.

## 2.2. Modelagem do Banco de Dados

Para desenvolvimento do sistema de crawling dos websites optamos por criar um Banco de Dados normalizado para que quando extraíssemos e salvassemos as informações desses websites, os dados já seriam salvos em uma estrutura normalizada. Portanto foram desenvolvidos o Modelo Conceitual, Modelo Relacional e Modelo Lógico do Banco de Dados.

Para o Sistema de Banco de Dados Relacional foi escolhido o PostgreSQL, sob licença BSD, disponível para diversos sistemas operacionais, por possibilitar o uso de recursos para controle de transação, que segue o modelo ACID, e por possibilitar orientação a objeto permitindo o uso de herança entre tabelas. A orientação a objeto poderia ser utilizada para tabelas com relacionamento de generalização/especialização.

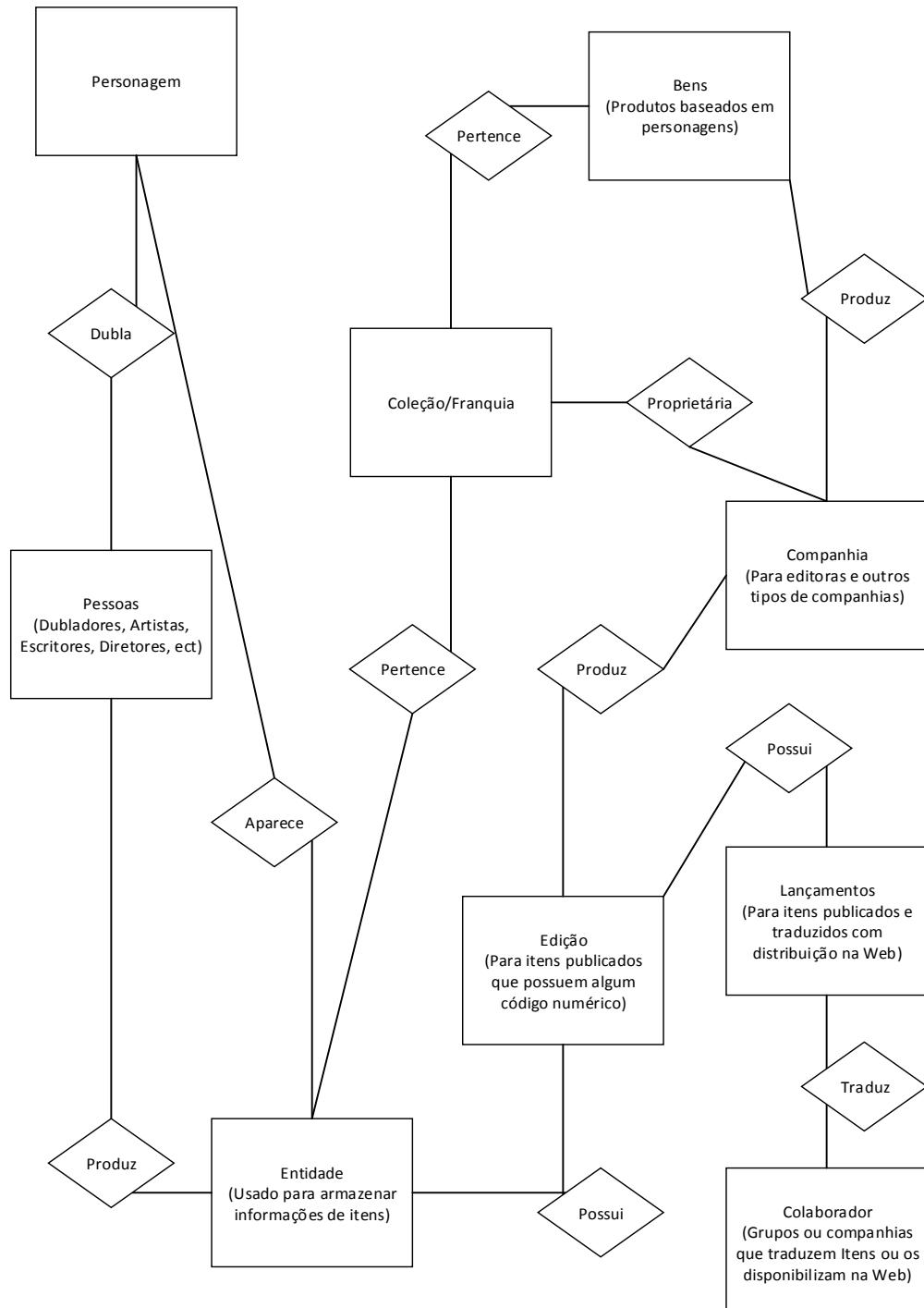
### 2.2.1. Modelo Conceitual

Com o estudo do conteúdo dos websites escolhidos, o conceito inicial de armazenar dados sobre mangás, animes e Light Novels resultou em um Modelo Conceitual expandido que possibilita a inserção de diversos dados relacionadas a franquias como músicas, softwares, vídeos e livros, além de permitir o cadastro de qualquer pessoa e empresa envolvida na produção de algum produto.

Com a expansão do tipo de informação a ser salva no banco de dados, a criação

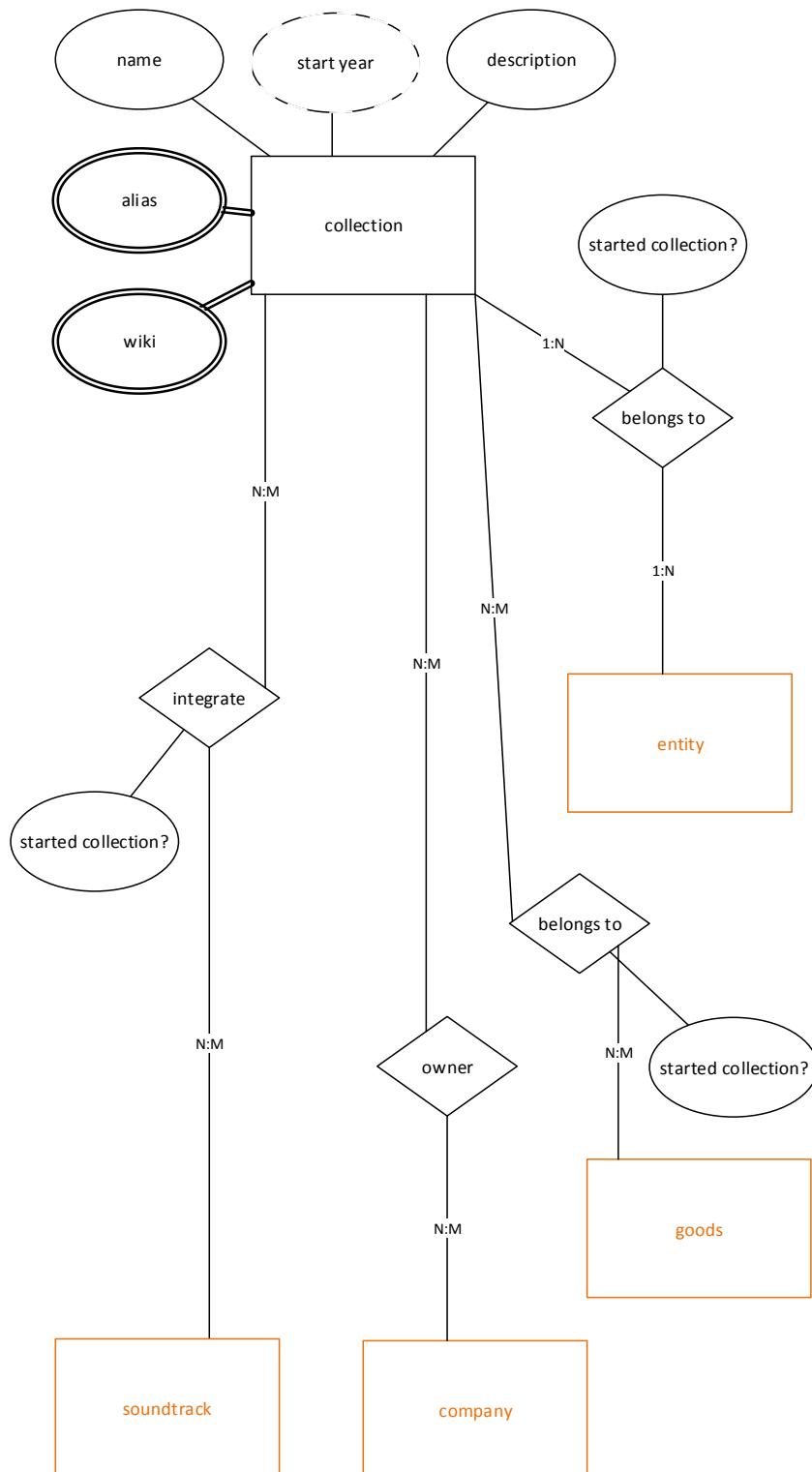
do modelo conceitual se tornou complexa: além do tradicional uso de atributos e relacionamentos foram utilizados conceito de especialização/generalização e associação.

Como a estrutura do Modelo Conceitual é complexa dividimos sua ilustração em partes. Para melhor compreensão, antes de apresentarmos cada parte mostraremos um diagrama mais simples abrangendo as principais entidades do Modelo Conceitual:

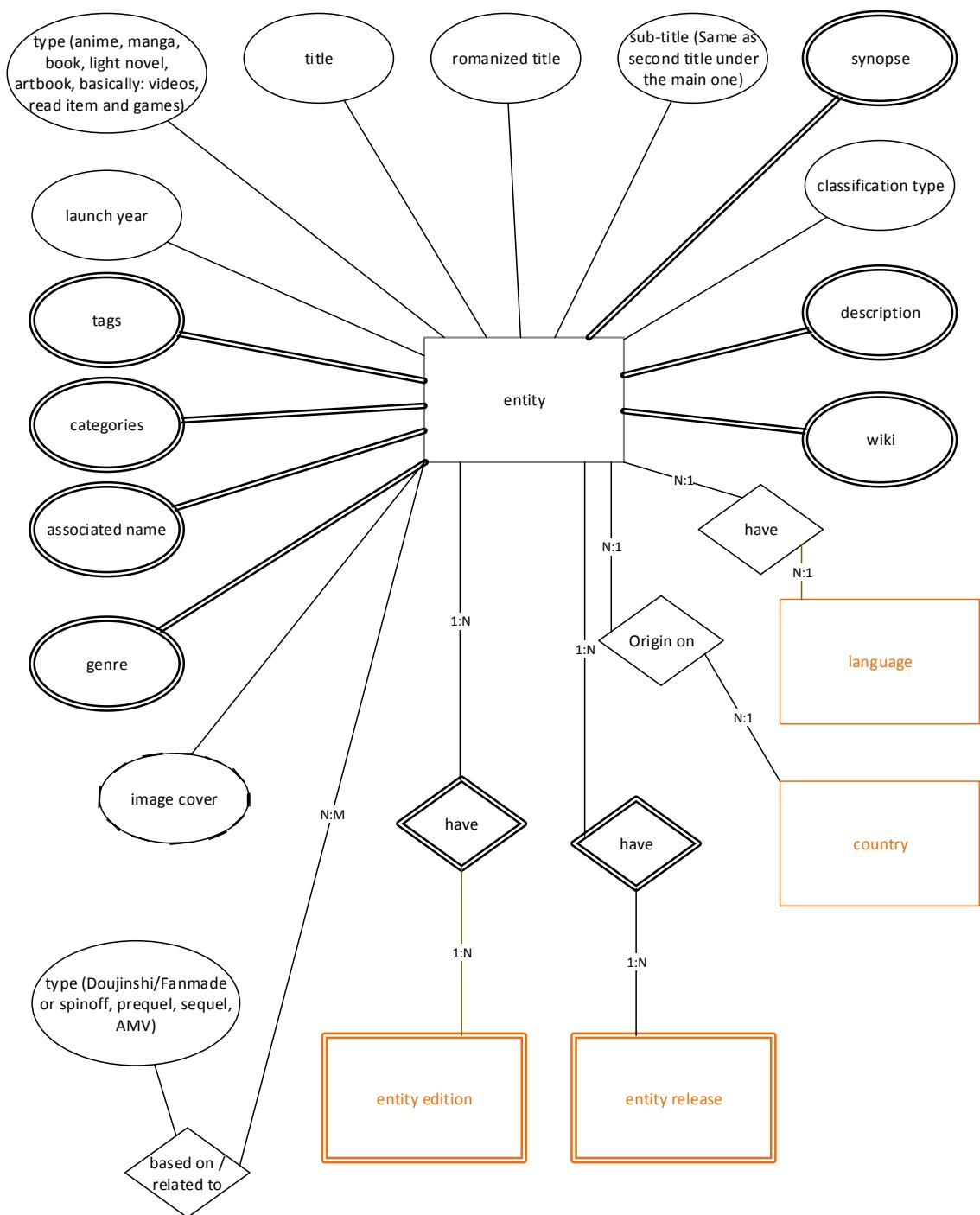


**Figura 8. Modelo Conceitual resumido com os nomes das principais entidades em português. No Modelo Conceitual detalhado e na implementação do Modelo Lógico foram utilizados textos em inglês.**

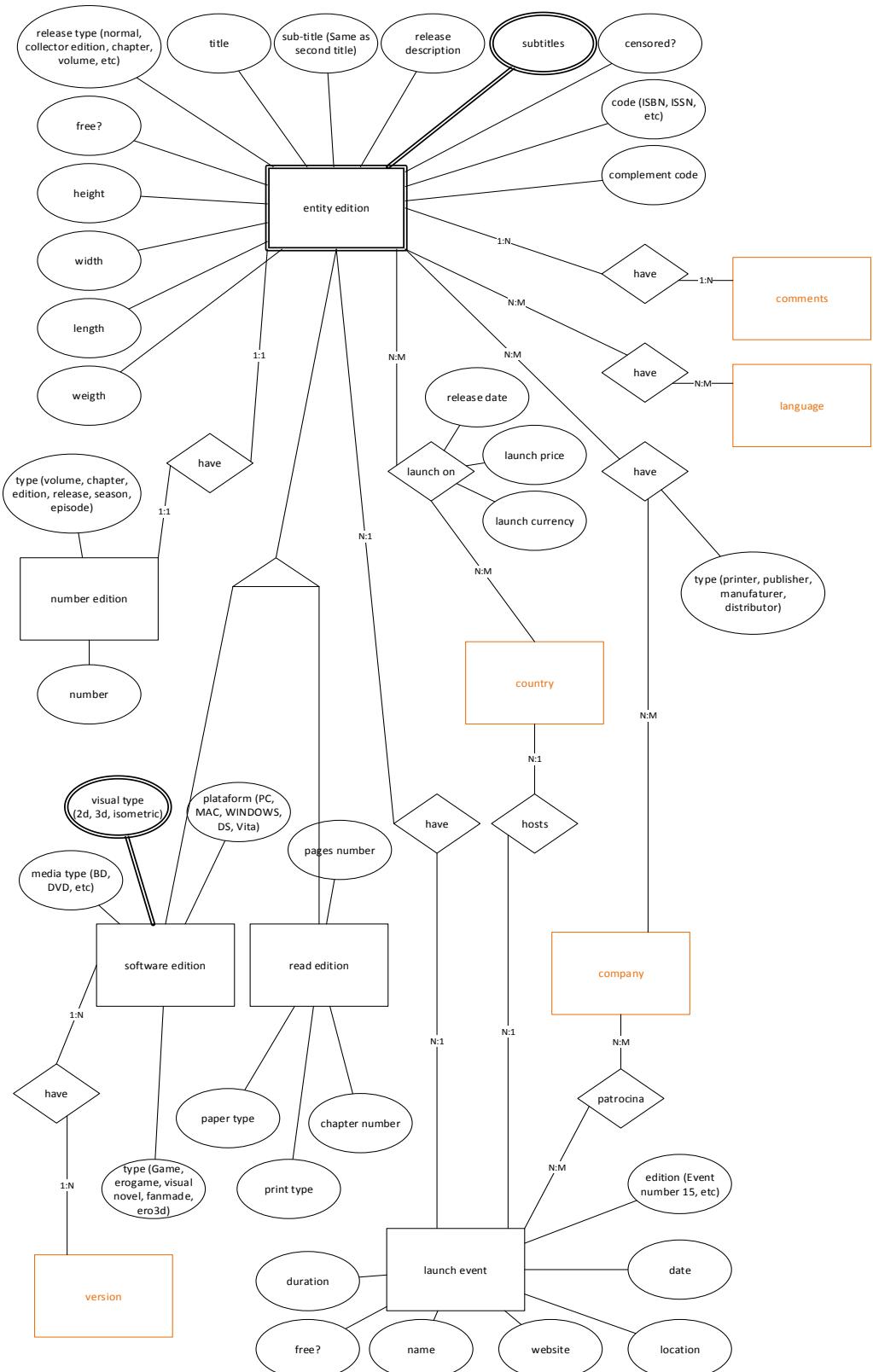
A seguir podem ser observados as entidades e seus relacionamentos de forma mais detalhada. As entidades na cor laranja representam entidades que serão detalhadas mais a frente.



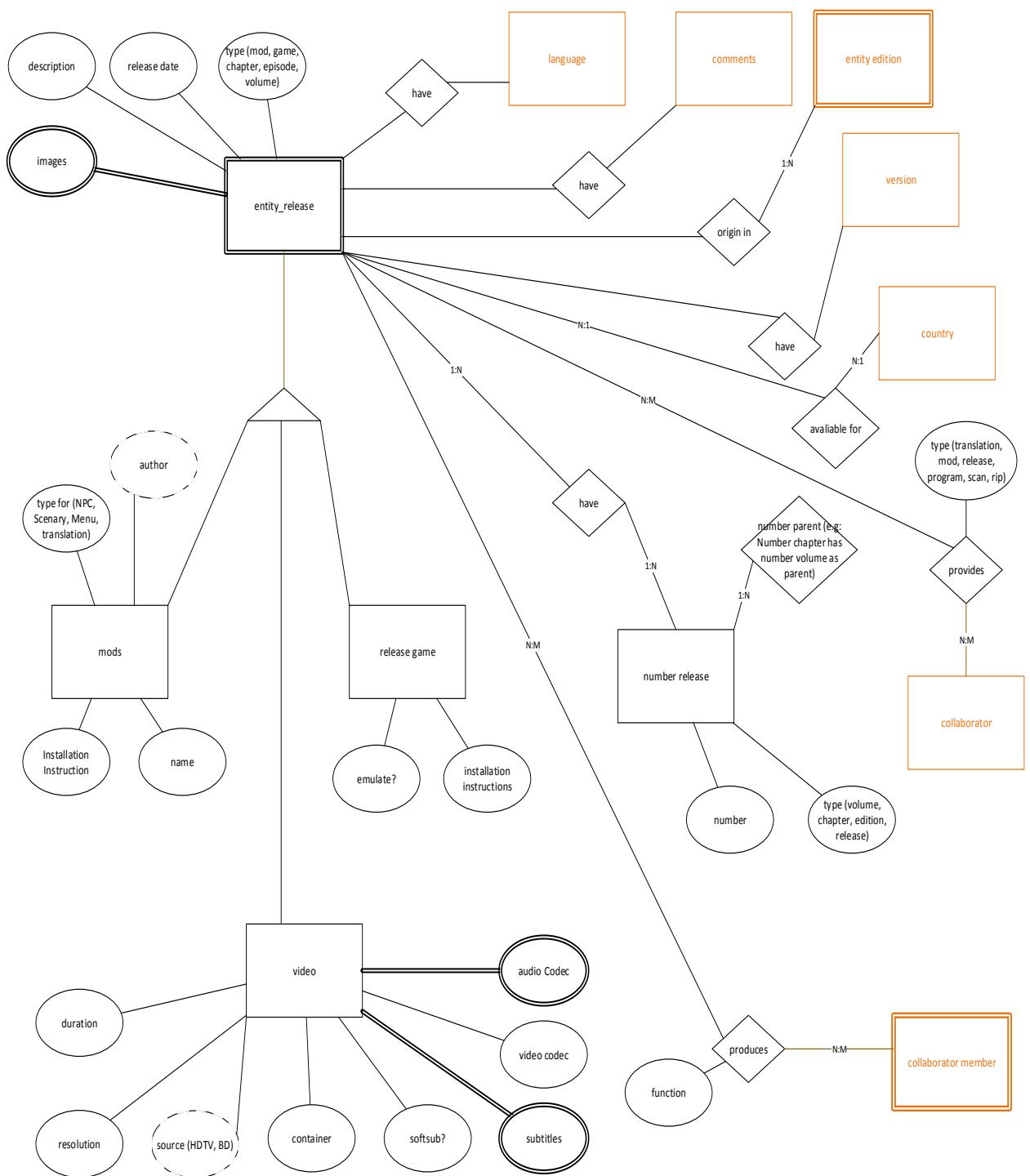
**Figura 9.** *Collection* é a entidade responsável por armazenar informações de franquias.



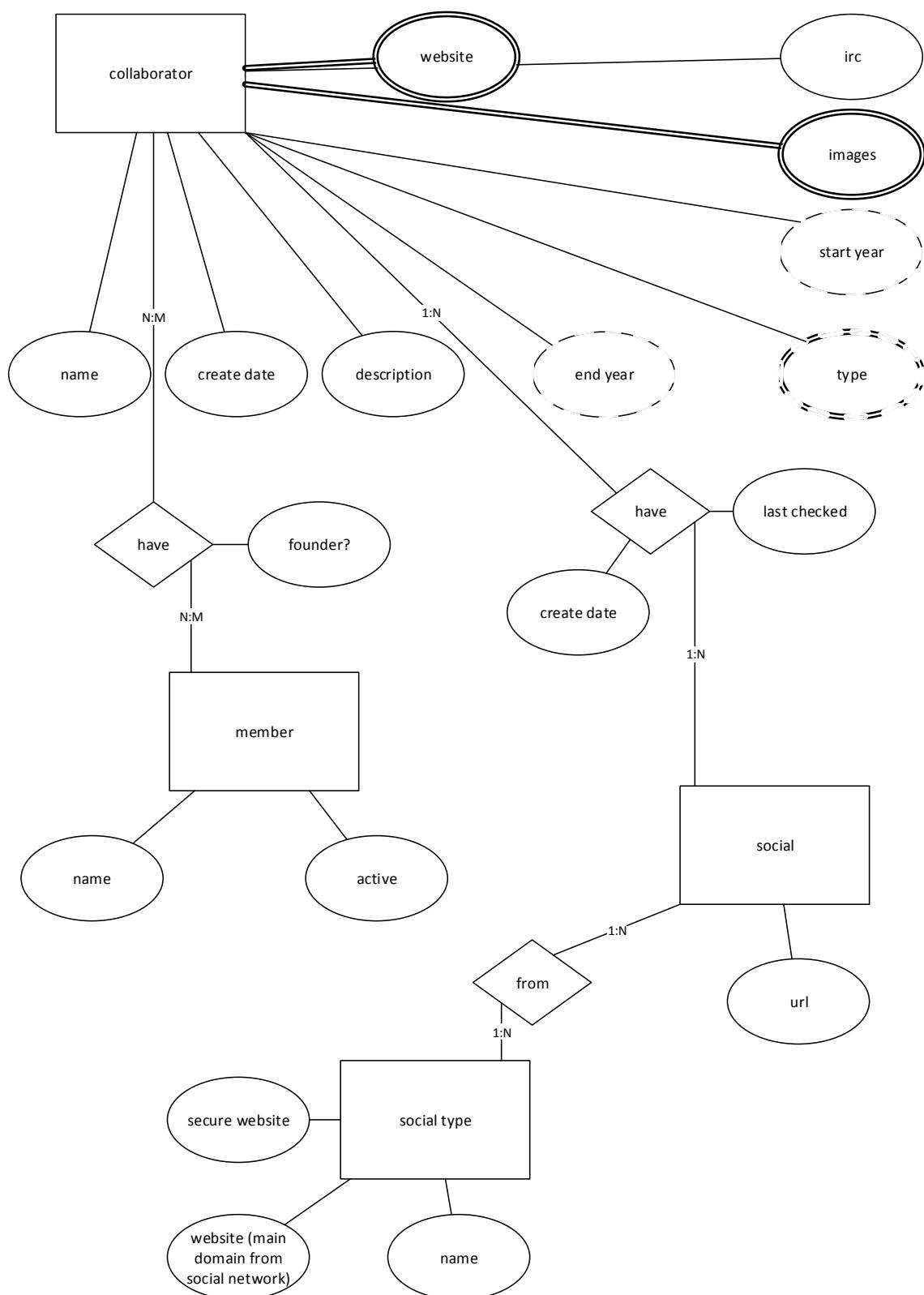
**Figura 10.** *Entity* é a entidade responsável por armazenar diversos tipos de conteúdo como vídeos, livros e jogos.



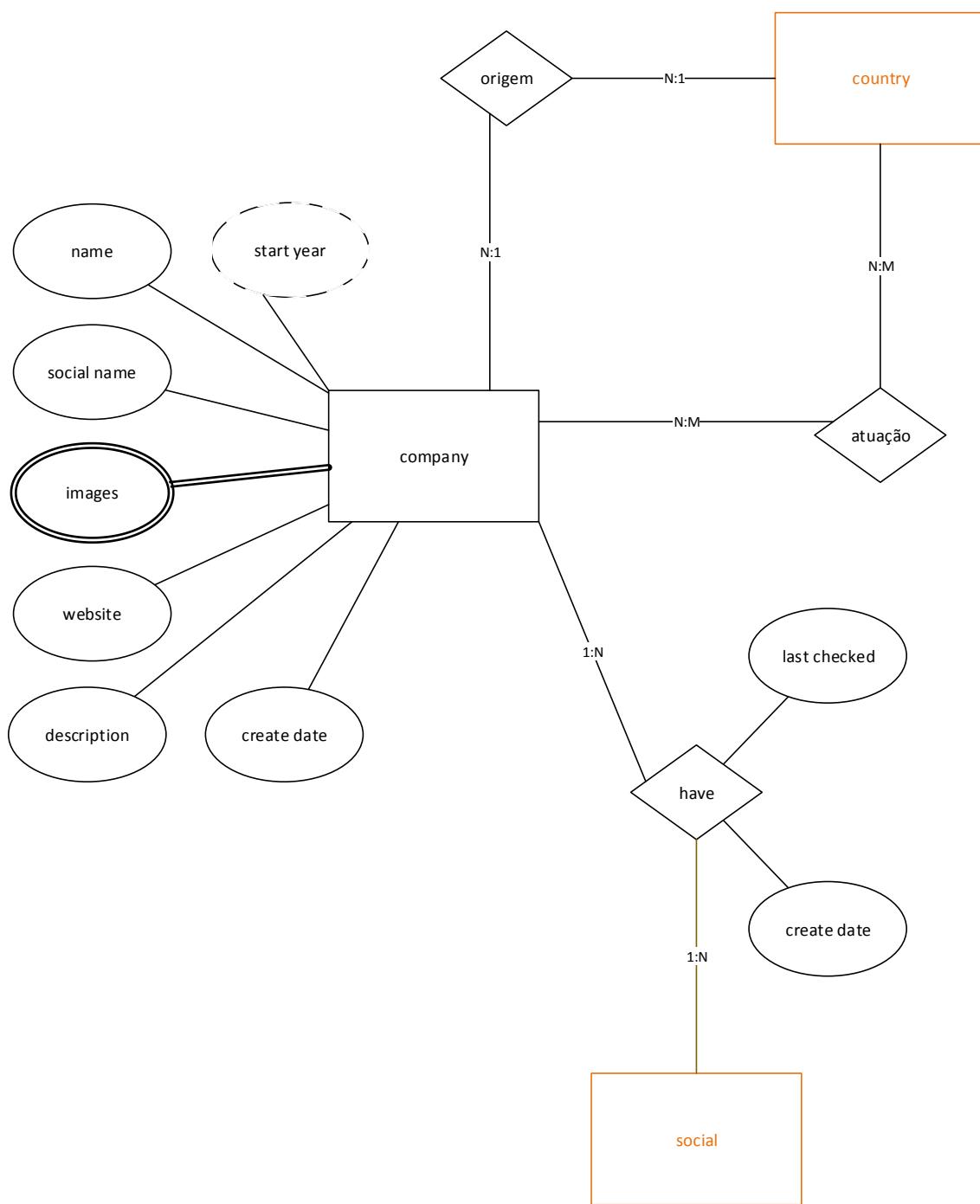
**Figura 11.** *Edition* é a entidade responsável por armazenar informações de itens armazenados em *Entity* que possuem publicação física. A entidade *Edition* se especializa em *Software Edition* para armazenamento de informações específicas a softwares e *Read Edition* para armazenamento de informações de livros e revistas



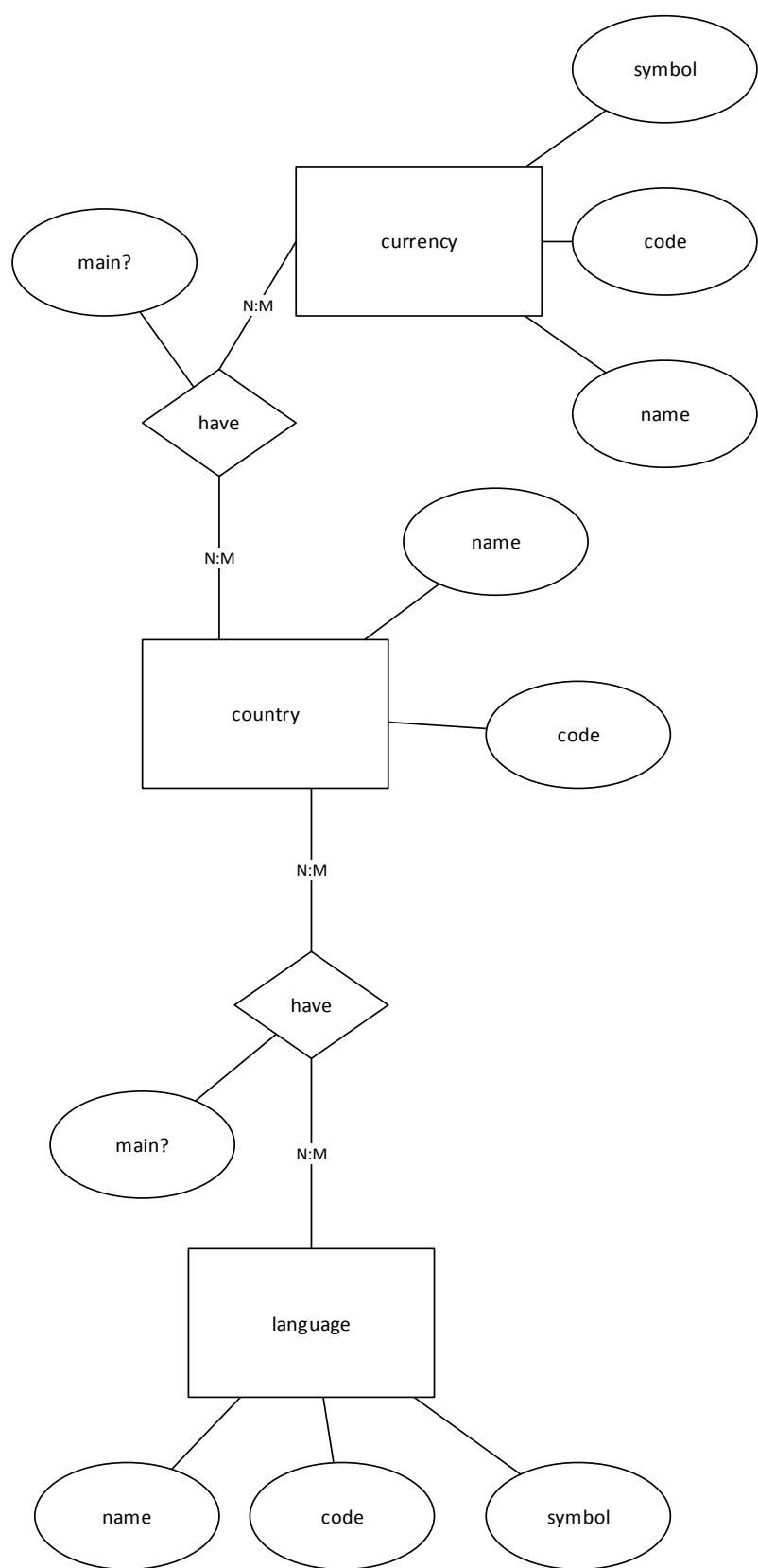
**Figura 12. Entidades responsáveis pelo armazenamento de conteúdo disponibilizado na web como modificações de jogos, conhecidos pela abreviação Mod, de traduções não oficiais de conteúdo ainda não licenciado fora do Japão e de distribuições de conteúdos, legalmente, através da Web.**



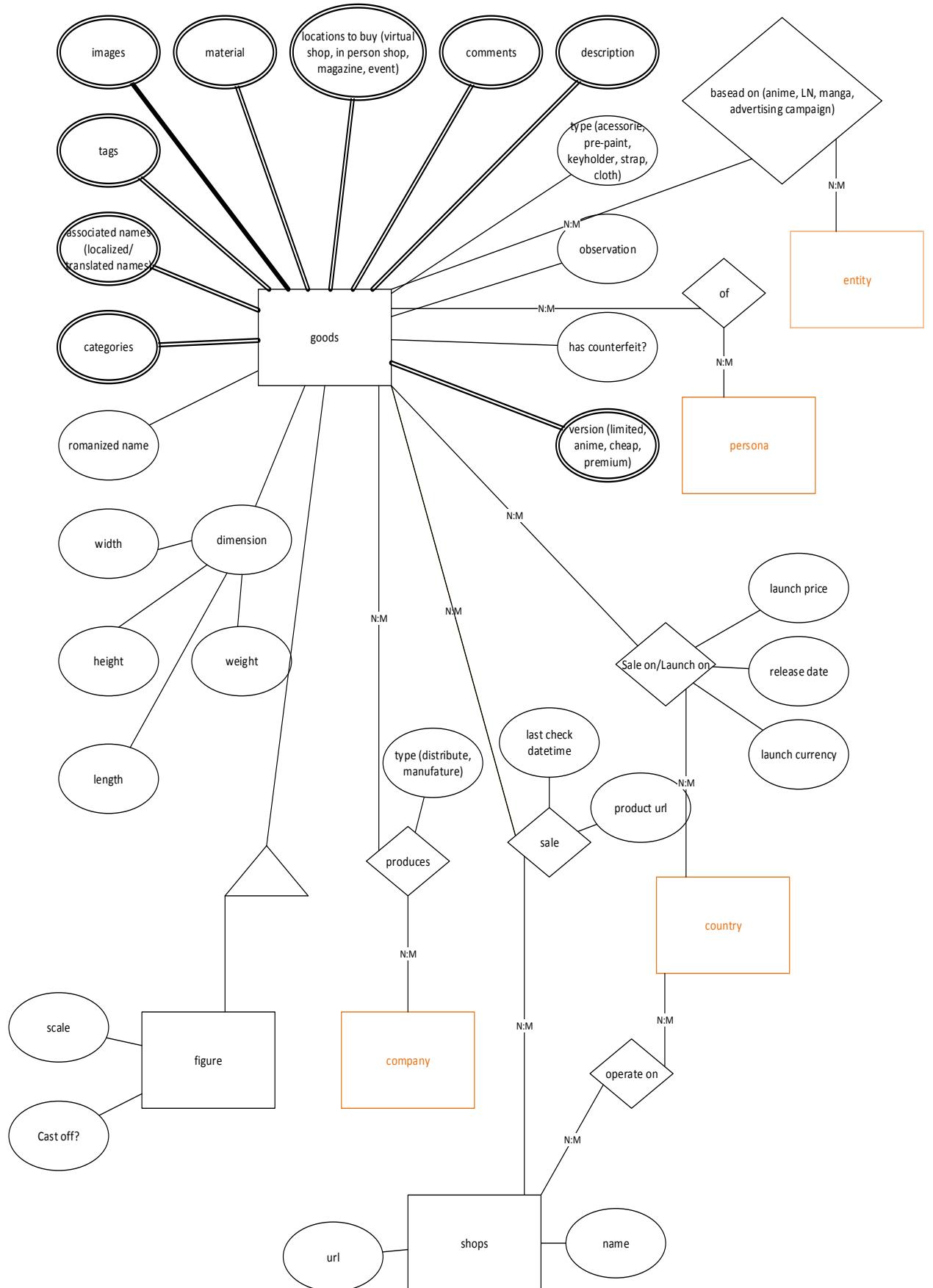
**Figura 13. Entidades com informações sobre grupos de tradução ou distribuidores de conteúdo digital, seus membros e suas redes sociais.**



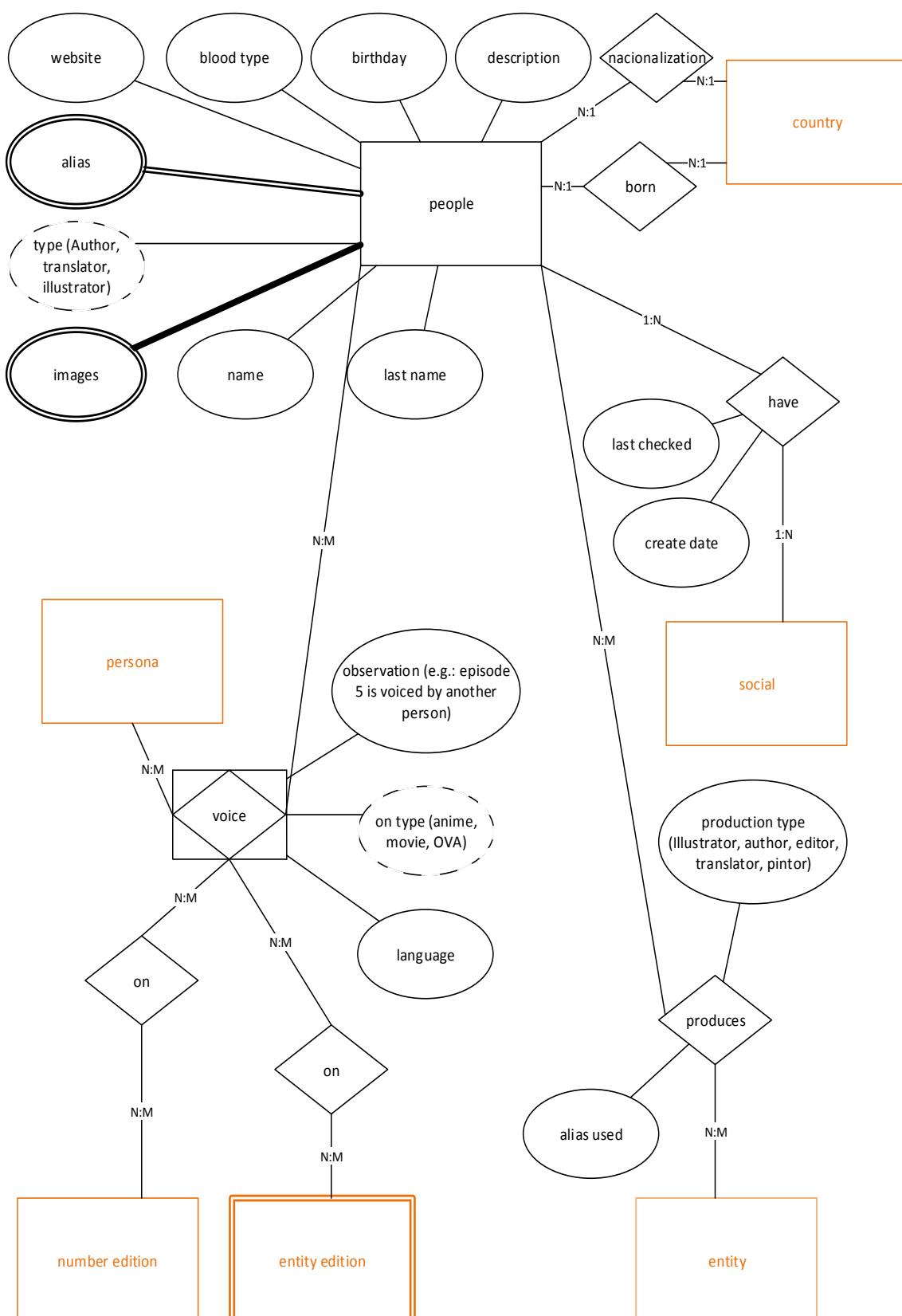
**Figura 14.** *Company* é a entidade responsável por armazenar informações de diversas empresas envolvidas na produção de itens. Atributo *images* é utilizado para armazenar apenas a referência das imagens, uma vez que imagens não são armazenadas diretamente no banco de dados.



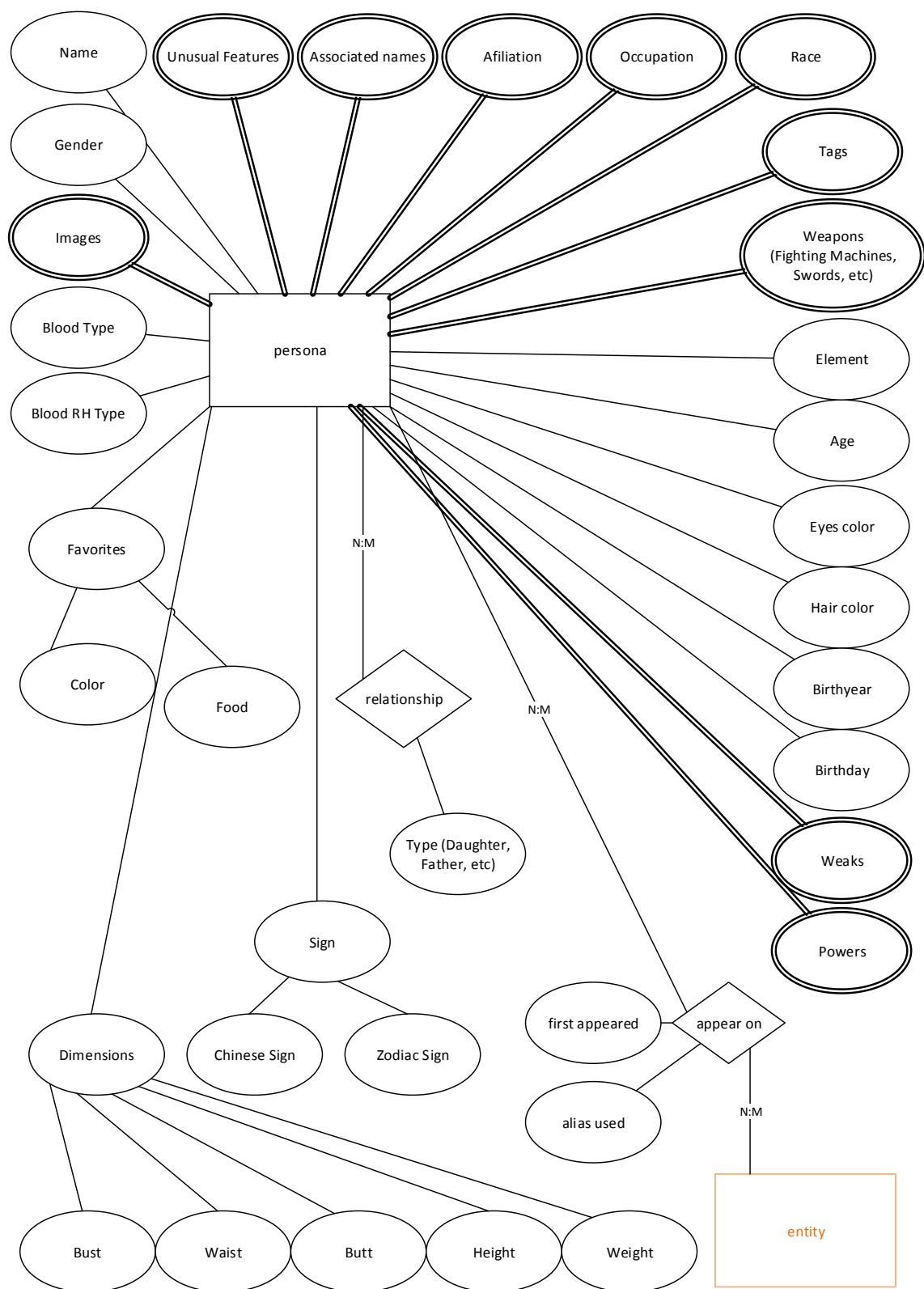
**Figura 15.** Entidades responsáveis pelo armazenamento de informações de países, seus idiomas e suas moedas.



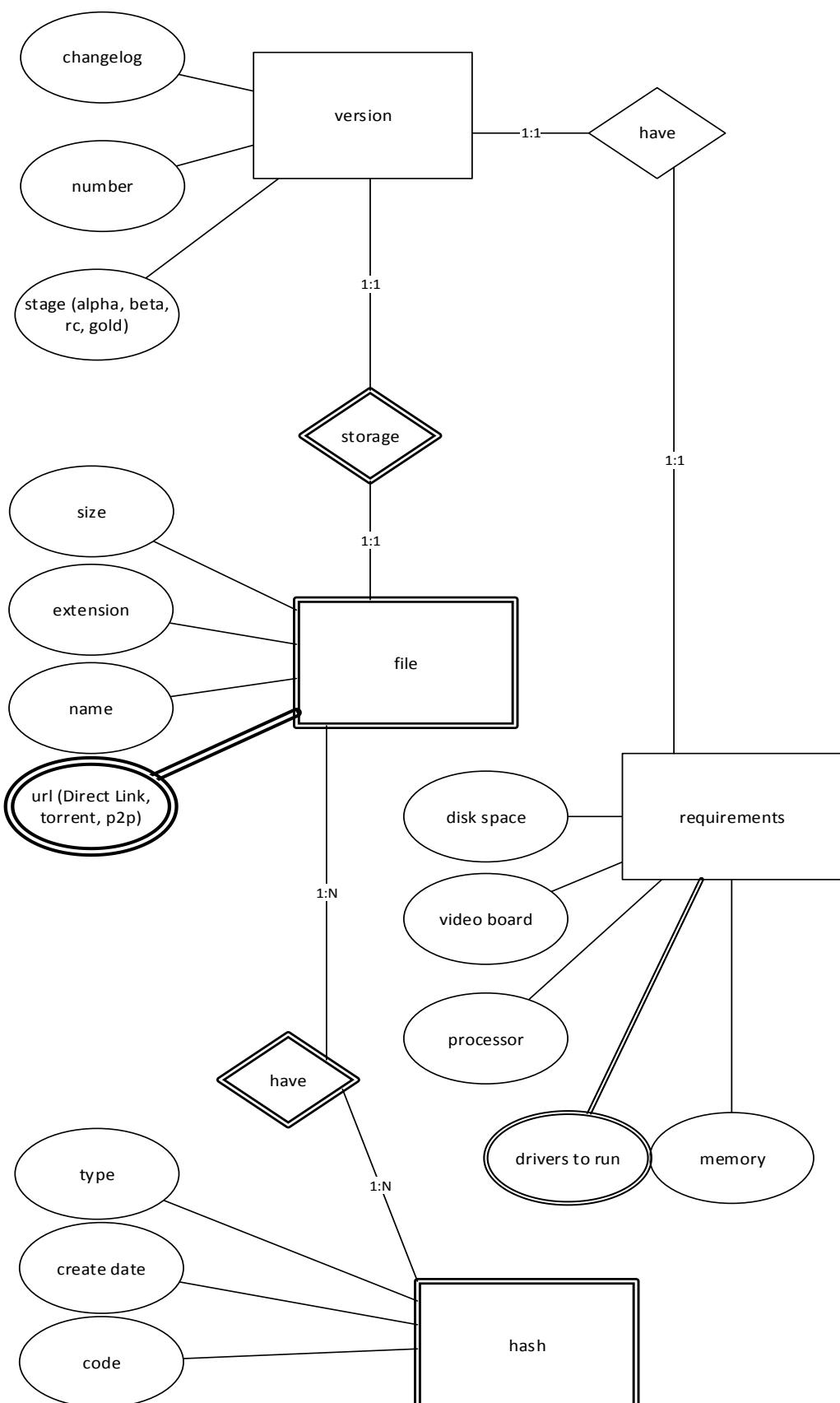
**Figura 16. Entidade *Goods*, responsável por armazenar dados de produtos, se especializa na entidade *Figure*, responsável por armazenar figuras de ação. Cast-off é o termo utilizado, entre colecionadores, para se referir a opção de remoção de roupa que algumas figuras de ação oferecem.**



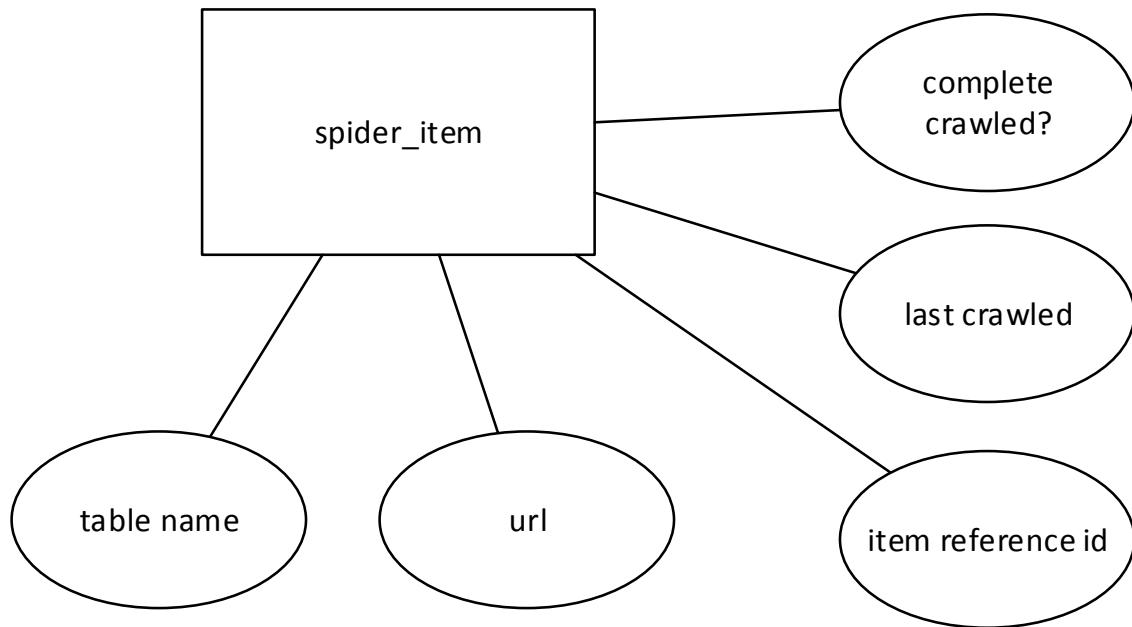
**Figura 17. Entidades responsáveis pelo armazenamento de informações de pessoas envolvidas na produção de itens. Entre o relacionamento de pessoas com personagens há associação utilizada para indicar em quais edições e episódios (*number edition*) o dublador participou. Há situações onde o dublador oficial não pode dublar um episódio ou uma lista de episódios por motivos fora de seu controle.**



**Figura 18. Entidade responsável por armazena personagens e suas características.**



**Figura 19. Entidades responsáveis pelo armazenamento de informações de versões existentes de softwares, animações e livros e seus respectivos arquivos disponibilizados na Web.**



**Figura 20.** Entidade responsável pelo armazenamento das URLs que tiveram informações extraídas e salvas no banco de dados. Essa entidade não é utilizada para verificar se uma URL já foi visitada durante uma execução do Crawler, isso é feito no próprio Crawler. Essa entidade serve para garantir que cada URL não salve informações duplicadas em multiplas execuções do Crawler, por isso é salvo a referência do ID e a tabela em que a informação foi salva. É permitido que alguns dados duplicados sejam salvos, por exemplo títulos, para casos em que itens diferentes possuam o mesmo nome.

### 2.2.2. Modelo Relacional

Após a criação do Modelo Conceitual foram criadas as tabelas e relacionamentos do Modelo Relacional seguindo as três Formas Normais, separando portanto atributos multi-valorados, não relacionados a totalidade da chave primária em tabelas próprias quando necessário.

Também foram criadas chaves-primárias não naturais, como índices numéricos que são auto incrementados na inserção de conteúdo, e foram definidos valores padrão para atributos que armazenam Data e Hora.

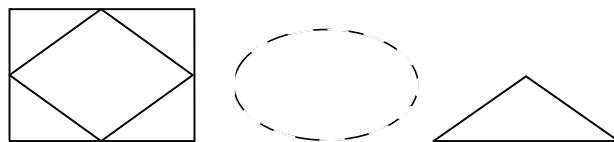
Para as entidades que possuem mais de um tipo de título ou descrição, como *Entity*, *Persona* e *People* que possuem o título principal e título, foram criadas entidades para armazenamento desses títulos ou descrições com um atributo que identifica também o idioma do texto.

### 2.2.3. Ferramentas Utilizadas para Modelagem

#### Modelo Conceitual

Para a modelagem do Banco de Dados e criação do Modelo Conceitual foi escolhido o Microsoft Visio que permite a geração de diversos tipos de diagramas, possibilitando a criação de novos itens se necessários para uso nos diagramas. Alguns dos itens como

Associação e Atributos Compostos não estão presentes por padrão na modelagem de dados que vem com o Microsoft Visio e tiveram que ser criadas.



**Figura 21. Objetos que precisaram ser criados: associação (a), atributo derivado (b) e generalização/especialização (c).**

## Modelo Relacional

Para o Modelo Relacional foi utilizado o DBDesigner na sua 4<sup>a</sup> versão, que permite a migração do Modelo Relacional para código SQL do MySQL. DBDesign não oferece suporte ao PostgreSQL sendo necessário para criação do Modelo Lógico algumas alterações no arquivo SQL.

Para um Modelo Relacional extenso o DBDesigner porém se mostrou limitado ao não permitir o aumento na área útil em que as entidades poderiam ser adicionadas. Para uma modelagem extensa como a do banco de dados desse projeto essa limitação dificultou a organização pela falta de espaço fazendo com que as entidades ficassem muito próximas umas das outras.

Além dessa limitação a execução da máquina virtual do JAVA (JVM) para o DB Design apresenta problemas com a hibernação no Sistema Operacional Windows, quando se retorna de uma hibernação, com a JVM ativa, ao tentar salvar ou exportar um arquivo no DBDesign o programa trava sendo necessário reniciar o computador para voltar a utilizar essas opções. Portanto não é recomendado o uso do DB Design para um modelo relacional extenso.

## Modelo Lógico

Para execução do Modelo Lógico foi utilizado o PostgreSQL 9.3 e o PgAdmin III para o sistema operacional Windows 8.

Por padrão o usuário inicial no PgAdmin III é o postgres, mas para execução de queries de seleção, inserção e atualização foi criado outro usuário para uso no nosso sistema crawler.

### 2.2.4. Detalhes da Implementação do Modelo Lógico

Ao exportar o arquivo SQL do DBDesigner é criado código SQL especificamente para MySQL, portanto para usarmos no PostgreSQL esse código precisou ser alterado.

O código SQL resultante do DB Design possui a criação de índices e chaves estrangeiras inclusas na criação da própria tabela. Apesar de ser compatível com o PostgreSQL resolvemos separar esse código em outros arquivos para uma melhor organização.

No MySQL chave-primárias sequenciais são definidas como valor numérico com

propriedade *auto\_increment*. Como *auto\_increment* não está disponível para o PostgreSQL usamos o equivalente **SERIAL**.

Código MySQL:

```
id integer primary key auto_increment
```

Equivalente em PostgreSQL:

```
id serial
```

Outra propriedade presente no código MySQL que não está disponível em PostgreSQL é o **DATETIME**, usado para armazenar data e hora. Uma alternativa adotada foi utilizar **timestamp**, que também armazena data e hora, porém em microsegundos desde o epoch Unix (01/01/1970 00:00). Para inserção do timestamp, além de ser aceito o tempo em microsegundos, se fornecida string concatenada de data e hora no padrão 'AAAA-MM-DD HH:MM:SS' o PostgreSQL fará a conversão automaticamente para microsegundos.

No PostgreSQL há dois tipos de timestamp: com fuso horário e sem fuso horário. Para salvar timestamps que não possuam um fuso horário ainda determinado, é indicado o uso de timestamp sem fuso horário. Nesse projeto foi escolhido usar timestamp com fuso horário para indicar quando alguns dados foram inseridos.

Código MySQL:

```
data_cadastro DATETIME NOT NULL DEFAULT now()
```

Alternativa em PostgreSQL:

```
data_cadastro timestamp without time zone NOT NULL DEFAULT now()
```

Um recurso presente em PostgreSQL é a herança entre tabelas, que permitem que tabelas herdem atributos não-chave de outras tabelas. Usamos esse recurso em entidades que possuem especialização, como as entidades *goods* e *edition*. Como herança não é um recurso presente no MySQL durante a criação do Modelo Relacional no DB Design foi utilizado relacionamento de cardinalidade 1:1 entre as tabelas que sofrem especialização/generalização.

O PostgreSQL em sua versão estável mais recente, 9.3, ainda não oferece criação automatizada de relacionamentos e unicidade entre tabelas pais e descendentes quando se indica uma herança na tabela descendente. Atributos chaves-estrangeira usados na tabela pai, que também devem estar presentes na tabela descendente, precisam ser inclusos manualmente.

Código para criação de tabela com herança em PostgreSQL:

```
CREATE TABLE figure (
    ...
) INHERITS (goods);
```

Uma pesquisa no banco de dados PostgreSQL efetuada em uma tabela pai automaticamente inclui resultados presentes nas tabelas descendentes. Para evitar esse com-

portamento é necessário indicar que se deseja procurar apenas na tabela pai. Isso pode ser indicado utilizando **ONLY** antes do nome da tabela no código SQL de consulta.

Código de consulta em PostgreSQL com **ONLY**:

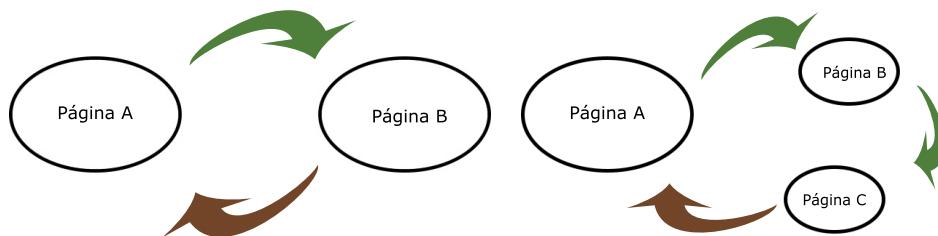
```
SELECT * FROM ONLY goods;
```

### 2.3. Crawler

Um crawler é um programa que visita websites, segue links e extrai dados de páginas específicas. Um crawler deve a partir de uma URL inicial seguir recursivamente todas as URLs acessíveis a partir da inicial. Critérios podem ser utilizados para indicar quais URLs seguir, como por exemplo o critério de considerar apenas o domínio atual do site, excluindo websites externos de publicidade e redes sociais.

Na execução inicial o Crawler faz o download do código HTML da página da URL inicial e extrai a informação "href" das tags `<a>` presente na página e as coloca numa lista de processamento usada na análise que determina se a URL deve ser seguida, extraída ou removida da lista.

Para evitar o problema de loops infinitos, por exemplo, quando uma página A possui um link para uma página B e a página B possui link para a página A, deve-se criar uma lista de URLs já visitados para verificação. Esse problema não necessariamente ocorre em uma página imediata a outra, podendo ocorrer mais tarde durante a execução do crawler.



**Figura 22. Figura (a) exibe o problema ocorrendo já na página seguinte.**

Neste projeto optou-se pelo uso de uma biblioteca de crawling que permite o seguimento de URL de páginas e extração de informações sem ter que se preocupar em programar explicitamente Request e Download de páginas da Web.

A Biblioteca escolhida foi o Scrapy, na versão 0.24.4, desenvolvida em Python, que possibilita a criação da lógica de URLs a serem seguidas usando regras conhecidas como *Rules*. As *Rules* são classes com parâmetros que definem se as URLs podem ou não ser inseridas na lista de processamento. Esses parâmetros recebem expressões regulares que serão usadas na análise das URLs. Nas *Rules* também é definido se a URL deve ter seu conteúdo HTML processado pelo parse ou se deve ter apenas as informações href das tags `<a>` extraídas.

Como os websites a serem utilizados possuem conteúdo estático sem uso de Ajax, o Scrapy atende as necessidades básicas de download e parseamento dos websites. Se for necessário extraer informações dinamicamente geradas por AJAX pode ser utilizado o

Scrapy com extensão para Firefox ou outra biblioteca de crawling que permite o download do código-fonte da página como visualizado pelos browser, como a biblioteca Selenium.

Scrapy, assim como Python, está disponível para Windows e Linux. Nesse projeto foi utilizado a versão para Windows de 64bits.

### 2.3.1. Links Duplicados

Para se evitar o problema de loops durante o crawling, URLs já processadas precisam ser ignoradas durante a extração de URLs de novas páginas.

Adicionar uma URL já processada a uma lista usada para verificação de URLs já visitadas evita esse problema, porém há casos em que duas URLs com a mesma quantidade de parametros, mas que possuem ordenações diferentes, levam a uma mesma página. A fim de evitar re-processar páginas iguais simplesmente pela ordem dos parâmetros estarem diferentes a biblioteca Scrapy não salva as URLs em si, e sim uma representação das URLs mencionadas como "impressão digital" na documentação do Scrapy.

Para gerar as "impressões digitais" a biblioteca Scrapy reorganiza os parâmetros de uma URL por ordem alfabética e gera a partir da URL resultante o MD5.

Exemplo de URLs que resultam na mesma impressão digital:

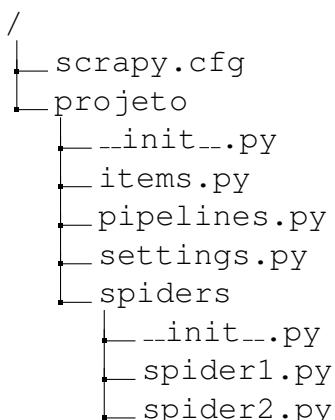
<https://www.mangaupdates.com/series.html?page=2&letter=AF>

<https://www.mangaupdates.com/series.html?letter=AF&page=2>

### 2.3.2. Utilizando Scrapy

Scrapy é uma biblioteca com todos os recursos necessários para extrair dados de um website, utilizando *Requests* para download do código-fonte de websites e *Parse* para processamento desse código-fonte. Os Requests e Parses são realizados por meio de classes conhecidas como *Spider*.

Um novo projeto Scrapy pode ser iniciado pela linha de comando e ao ser iniciado será criado uma estrutura pronta de pastas e arquivos padrões com algumas configurações pré-determinadas.



Para a visitação de URLs e extração de dados dos websites a biblioteca Scrapy utiliza as classes *Spider* armazenadas na pasta "spiders". Essas classes devem herdar de pelo menos um tipo de spider disponível no Scrapy, cada um com sua implementação única.

A classe que escolhemos herdar para criar nossos spiders é a *CrawlSpider* que implementa o método *parse*, método esse que utiliza as regras definidas para ou seguir uma URL ou efetuar processamento do código-fonte da URL. Quando definimos as regras, em um objeto iterável com nome "rules" na classe, indicamos por meio de expressão regular as páginas que podem ser seguidas e as páginas que devem ter seu conteúdo extraído, parseado e enviado para um método callback. Pode apenas haver um método callback para extração de dados nos spiders que herdam *CrawlSpider*.

Nossa classe *spider* para ser corretamente executada pela biblioteca Scrapy deve informar a propriedade nome do "spider", domínios permitidos para extração de dados sem o schema "http", as URLs iniciais e as regras de visitação e extração de URL.

Ao criarmos o método callback podemos usá-lo de pelo menos dois modos:

1. Usá-lo para armazenar as informações extraídas em objetos conhecidos como *items* para serem depois interpretados por diversos métodos *Pipeline* que verificarão se os *items* estão dentro de uma qualificação lógica adequada para serem salvos em Banco de Dados ou salvos em arquivos Json (Esse é o método recomendado pelo Scrapy).
2. Usá-lo diretamente para salvar as informações sem a necessidade de criação de objetos *items*.

Adotamos esse último método na qual as URLs extraídas e parseadas possuem seus dados normalizados para serem salvos no banco de dados no próprio método *callback*.

### 2.3.3. Implementando acesso ao Banco de Dados

Para acesso ao banco de dados PostgreSQL, em Python, foi utilizado a biblioteca Psycopg2. A biblioteca oferece métodos para conexão com o banco de dados, para execução de código SQL e para controle de transações.

A partir desses métodos básicos criamos uma classe de conexão ao banco de dados, com métodos para inserção e para atualização de dados que manipulem as informações nas tabelas mantendo a integridade lógica dos dados.

Embora o Banco de Dados seja responsável por manter a integridade quanto aos tipos de dados a serem inseridos e existência de chaves-primárias e estrangeiras, ele não verifica se a associação realizada é a pretendida. Por exemplo, se queremos associar o autor A ao Livro A e associamos por algum equívoco o autor B, isso é considerado uma falha na integridade lógica dos dados.

#### **2.3.4. Tratamento de Erros e Transactions**

Algumas das páginas processadas pelo *parse* podem ter variações na formatação HTML, que podem originar erros na integridade lógica do Banco de Dados. Assim usando o conceito de exceções, quando ocorre uma falha na inserção de algum conteúdo da página processada pelo *parse* todas as operações executadas com os dados da página são descartadas, usando *Rollback* quando uma Exception ocorre, e a URL da página e informações sobre o erro são inseridas em um arquivo de *Log* para análise posterior.

Somente após realização com sucesso, de todas as operações, é finalmente efetuado *Commit* na transação. Permitindo assim que apenas dados formatados e relacionados corretamente sejam salvos no Banco de Dados.

#### **2.3.5. Itens Relacionados**

Alguns dos websites escolhidos a terem seu conteúdo extraído nas páginas de detalhes de seus itens também fornecem informação de itens relacionados, listando esses itens em tags <a>. Esses itens podem ser livros, mangás ou figuras de ação. Cada página fornece informações que ajudam a determinar o tipo de item e em qual tabela será salvo.

Esses relacionamentos entre itens também são salvos no banco de dados desse projeto. Através das tags <a> podemos extrair a URL do item e consultando a tabela spider\_item do banco de dados podemos identificar se o item já foi salvo, precisando portanto apenas inserir o relacionamento nas tabelas adequadas. Se um item ainda não foi criado, um item com valores nulos é inserido no banco de dados, na tabela apropriada, a fim de se obter um id e seu id e sua URL são adicionados na tabela spider\_item.

A tabela spider\_item é consultada sempre que uma página é processada para ter suas informações extraídas e salvas no banco de dados. Se a página que estiver sendo processada foi localizada na tabela spider\_item o item é atualizado utilizando o id registrado ao invés de ser criado um novo item. Assim é mantido o relacionamento mesmo entre itens que ainda não tiveram suas URLs processadas pelo crawler.

#### **2.3.6. Definindo Franquias**

Para alcançarmos o objetivo de fazer uma visualização de dados com extração de informações relacionadas a franquias, precisávamos identificá-las e criá-las a partir do conteúdo salvo no Banco de Dados.

Escolhemos criar as franquias na execução do próprio Crawler, utilizando o nome dos itens e quando disponível os nomes dos itens relacionados.

Para a criação do nome das franquias é identificado o nome principal nos títulos dos itens, para isso removemos subtítulos, que podem ser identificados como parte posterior de um hífen ou parte dentro de parenteses, dos títulos. Com o nome principal podemos pesquisar por franquias na tabela collection que possuem o mesmo nome, ou criar uma nova se nenhuma franquia for localizada.

No título de um item abaixo temos um exemplo de título a esquerda e subtítulo a

direita:

Kidou Senshi Gundam Unicorn - Char second comming

Se um item possui algum item relacionado salvo no banco de dados, antes de criamos uma nova franquia caso nossa busca não retorne resultados é executado uma busca recursiva, utilizando o recurso WITH RECURSIVE do PostgreSQL, a fim de retornar o nome e franquia de todos os itens relacionados ao atual. Se uma franquia associada a um item é encontrada os demais itens ainda não associados a franquia são atualizados.

Se nenhuma franquia associada é encontrada será necessário criá-la. Se existem itens relacionados, o nome da franquia será criado usando os nomes em sequência mais comuns entre os nomes principais dos itens. Se não existem itens relacionados será utilizado apenas o nome principal do item atual na criação da franquia.

Nos títulos abaixo podemos identificar o nome mais comum entre os nomes principais, neste caso Fairy Tail:

Fairy Tail - Season 1

Fairy Tail Freezing - Season 1

Com esse método há possibilidade de associar a franquias itens com nomes semelhantes ou iguais que pertencem a franquias distintas. Como exemplo, existe o mangá Magician criado por Sarae Kim e existe também o mangá homônimo criado por Yuriko Matsukawa que possuem histórias diferentes e que não se passam no mesmo universo fictício. Para corrigir casos assim é necessária uma verificação manual da associação de franquias posterior a execução do crawler.

### 2.3.7. Determinando Idioma

Alguns dos website escolhidos além de fornecerem o título de itens em inglês também fornecem os títulos em outros idiomas como em japonês, em francês ou em português. Para que esses títulos fossem salvos no banco de dados foi necessário determinar o idioma a qual pertencem.

A biblioteca Langid, para Python, consegue sem a necessidade de treino prévio identificar até 97 idiomas. Langid oferece o método *classify* para identificar o idioma, que retorna o idioma e a probabilidade desse idioma estar correto em determinado texto. Embora a probabilidade de acerto quando se utiliza frases curtas esteja abaixo de 50%, para idiomas com caracteres distintos como o japonês ou coreano essa probabilidade aumenta e pode chegar até 99%.

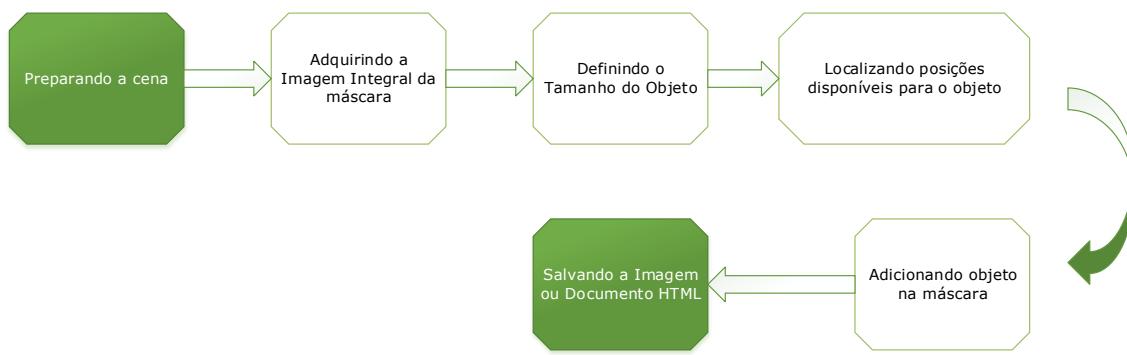
Portanto determinamos um valor mínimo para utilizar o idioma detectado pelo Langid. Quando a probabilidade retornada ultrapassava o valor de 70% usavamos o idioma identificado, mas quando esse valor não era alcançado utilizavamos o idioma padrão determinado. O idioma padrão para muitos itens foi determinado a partir do país de origem.

## 2.4. Visualização de Dados

Para visualização de dados extraímos do banco de dados textos e representamos a variação na quantidade entre os textos, exibindo-as como uma nuvem de bolhas ou uma

nuvem de palavras. Usamos a estrutura de dados Tabela de Área Somada, também conhecida em Visão Computacional como Imagem Integral, para identificação de espaços ocupados e realizamos uma distribuição aleatória na imagem de nossos objetos: as bolhas ou as palavras. Para esse fim utilizamos a biblioteca Numpy para manipulação de vetores multidimensionais, gerando assim a Tabela de Área Somada, e a biblioteca PIL para desenhar palavras, desenhar formas e salvar a imagem resultante.

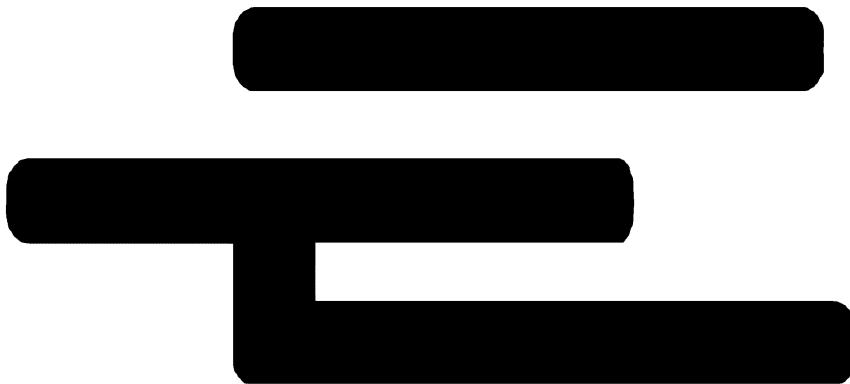
A imagem abaixo ilustra a sequência adotada para criação de nossas nuvens. Essas etapas serão melhor explicadas nos próximos tópicos.



**Figura 23. Etapas 2 a 4 são executadas múltiplas vezes enquanto ainda existir objetos a serem posicionados.**

#### 2.4.1. Preparando a Cena

Utilizando a biblioteca de processamento de imagem preparamos um fundo com tamanho determinado que usaremos para posicionar nossos objetos. Qualquer seguimento da Tabela de Área Somada com valor diferente de zero é considerado ocupado, assim podemos criar uma imagem com fundo totalmente preto, utilizando toda a área disponível, ou podemos criar uma área limitada com ilustrações ou textos resultando em uma máscara para nossa visualização. A cor preta é representada pelo valor 0 e ao converter a imagem preparada em um vetor multidimensional com Numpy podemos obter a matriz inicial que será utilizada para gerar a Tabela de Área Somada inicial.



**Figura 24. Imagem utilizada como máscara**

Para uso da Tabela de Área Somada é utilizado como máscara imagens na escala cinza, assim podemos trabalhar apenas com um valor variante para cada pixel da imagem.

#### 2.4.2. Obtendo a Tabela de Área Somada

A Imagem Integral é uma matriz obtida a partir de uma imagem que possui um único canal de cor. Cada valor na Tabela de Área Somada representa o produto resultante da soma de todos os valores anteriores da imagem.

Utilizando uma matriz inicial criamos a Tabela de Área Somada usando cumsum do Numpy. As dimensões da matriz inicial e a Tabela de Área Somada são as mesmas, o que as diferenciam são os valores de cada linha e coluna da matriz.

Imagen				Imagen Integral			
5	2	5	2				
3	6	3	6				
5	2	5	2				
3	6	3	6				

**Figura 25. Figura (a) representa a matriz formada por uma imagem com apenas um canal de cor. Figura (b) representa a Imagem Integral obtida a partir dos valores da figura (a).**

### 2.4.3. Definindo o Tamanho dos Objetos

Ao obtermos nossos objetos, palavras e círculos, do Banco de Dados também selecionamos informação numérica que representam um ranking entre os objetos. Alguns dos valores numéricos utilizados como ranking são: quantidade de itens da franquia, valor em reais para adquirir todos os item da franquia, volume mínimo em  $m^3$  necessário para armazenar todos os itens da franquia.

Para definir o tamanho dos objetos e manter uma proporção ao alor do ranking, usamos a razão do valor do ranking pelo logaritmo natural desse valor.

Como nossos resultados possuam um ranking na casa dos milhares, dividimos por 10 o resultado obtido para que não sejam gerados tamanhos demasiadamente grandes.

$$tamanho = \frac{c * ranking}{\ln(ranking + 10)} \quad (1)$$

Para evitar divisão por zero é somado o valor 10 ao número do rankeamento antes do cálculo do Logaritmo Natural.

Onde:

- $c$  – razão entre a área da cena e a quantidade de objetos
- $ranking$  – valor inteiro que representa o valor do objeto em relação aos outros

Para colocar os textos dentro dos círculos foi necessário determinar o tamanho que a fonte do texto deve possuir para não ultrapassar o diâmetro. Para isso foi utilizado o seguinte cálculo:

$$tamanho = \frac{d * 1.5}{c} \quad (2)$$

Onde:

- $d$  – diâmetro do círculo
- $c$  – quantidade de caracteres no texto

### 2.4.4. Determinando a posição dos Objetos

Para verificarmos se um espaço está disponível na cena para nossos objetos precisávamos verificar se já não existe outro objeto já posicionado no local.

Conseguimos verificar se o espaço está ocupado ou não calculando o valor na área que nosso objeto ocupa. Se o local em que pretendemos colocar nosso objeto estiver ocupado o resultado do cálculo da área será acima de zero.

Com o cálculo da área saberemos exatamente quanto espaço está sendo ocupado em determinada área e posição. E utilizando a Tabela de Área Somada conseguimos realizar o cálculo da área, em analise assintótica: em tempo constante O(1).

$$\text{área} = I(x, y) + I(x + i, y + j) - I(x + i, y) - I(x, y + j).$$

Onde:

- $I$  – representa a matriz da Tabela de Área Somada
- $i$  – representa o comprimento do objeto
- $j$  – representa a altura do objeto

Ao calcular a área de todos pixels da imagem, que serve de máscara, e adicionar em um vetor os valores que representam a linha e coluna da posição livre, poderíamos escolher randomicamente uma posição livre para adicionarmos nosso objeto.

Com uma posição escolhida informações do objeto, como tamanho e posição, são adicionados a um vetor utilizado posteriormente para geração da imagem final ou exportação para HTML.

Depois da escolha da posição e inserção do objeto na imagem máscara, a imagem é salva e é gerada uma nova Tabela de Área Somada a partir dessa nova imagem, para uso da verificação de espaço disponível da próxima palavra.

Quando um espaço disponível não for encontrado o objeto sofre redução de 1 pixel no seu tamanho e novamente é procurado por um espaço disponível.

#### **2.4.5. Gerando a Imagem Final ou Exportando para HTML**

Com a determinação do posicionamento dos objetos na cena, tamanhos e fontes de texto utilizadas além de gerar uma imagem final também podemos exportar nossa nuvem para HTML utilizando CSS para desenhar e posicionar nossos objetos dentro de uma tag `<div>`.

Para gerar a imagem final com nossos objetos posicionados utilizamos os métodos disponíveis na biblioteca PIL para inserção de texto e de elipse e depois salvamos a imagem final em um local determinado. A biblioteca PIL não oferece método para criação de círculo, mas podemos criá-los inserindo uma elipse que terá um raio constante, ou seja uma elipse com dimensões de um quadrado.

Para a criação do HTML consideramos que a posição dos objetos é relativa a uma tag `<div>` com elemento id definida como principal, as mesmas coordenadas [x, y] utilizadas no posicionamento em nossa imagem podem ser aplicadas no posicionamento na tag `<div> #principal`.

Geramos um documento HTML com nosso CSS adicionado na tag `<head>`, que permite uma melhor manipulação posterior de nossos objetos quando o HTML for salvo.

Cada objeto a ser posicionado deve estar dentro de uma tag `<div>` que contém classes que representam os tamanhos, formas e posições. Essas classes e seus estilos CSS são criados após a determinação de posição de todos os objetos.

Com o posicionamento dos objetos determinado, suas dimensões e outras características definidas, para gerarmos o documento HTML criamos métodos que armazenam em uma variável o texto com o estilo CSS utilizado, para nossos objetos, e em outra variável as tags `<div>` criadas. Ao finalizar o processamento de todas as palavras criamos um novo arquivo com extensão `html` e inserimos o conteúdo das duas variáveis formatadas corretamente nesse novo arquivo.

### **3. Resultados**

Com o uso de crawling foi possível obter uma quantidade considerável de dados. Um resultado mais detalhado pode ser visto a seguir:

Manga-Updates - Com o crawling do site <http://mangaupdates.com> foi possível obter 103.774 itens, classificados como Mangás, Light Novels e livros; 34.721 pessoas, que atuam como ilustradores ou escritores; e 1.181 editoras.

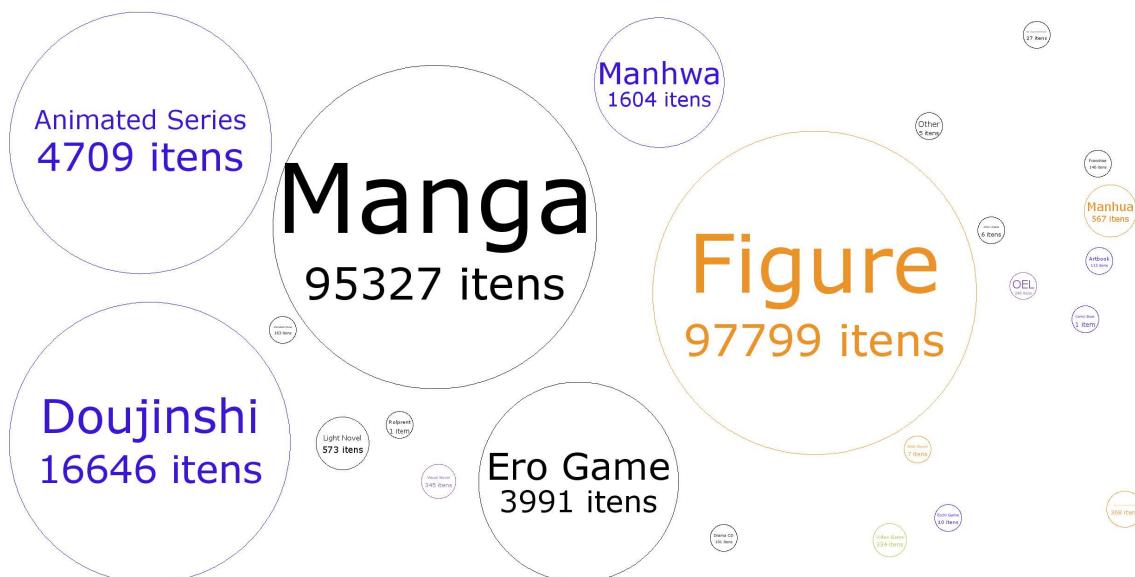
My Figure Collection - Com o crawling do site <http://myfigurecollection.net> foi possível obter 205.041 produtos, incluindo Figuras e 3.440 empresas envolvidas na produção desses produtos.

Anime Characters Database - Com o crawling do site <http://animecharactersdatabase.com/> foi possível obter informações de 70.304 personagens e 4.840 dubladores.

Com os resultados obtidos com o crawling foi possível criar 109.274 franquias e associá-las aos itens e aos produtos. Mas, dessas franquias apenas 1.497 possuem mais de um item. Também foi possível relacionar produtos com personagens.

Para obter esses resultados foi necessário a execução do crawler por três dias consecutivos, levando um dia para download e extração das informações de cada site.

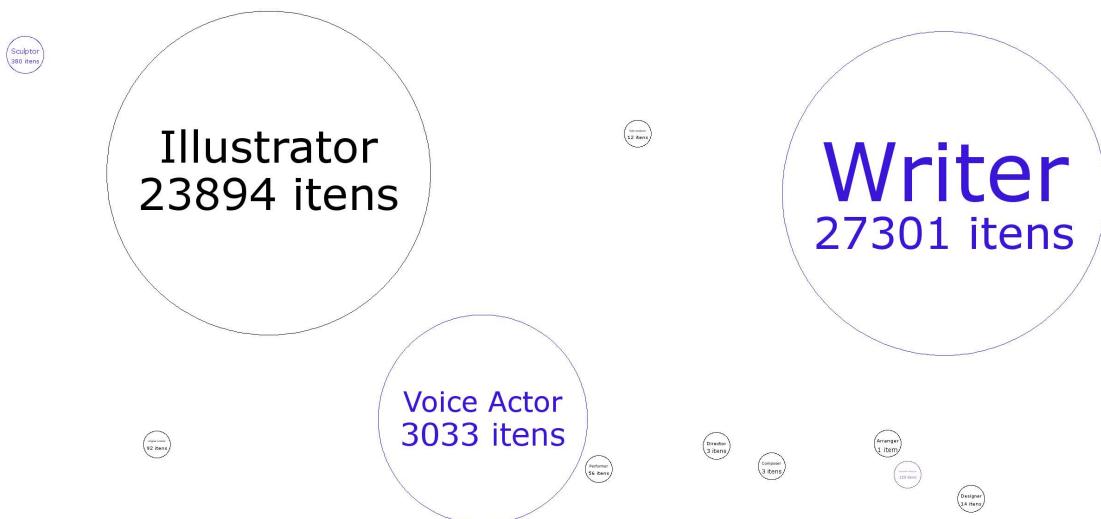
Com as informações armazenadas no Banco de Dados foi possível criar várias visualizações seguindo os dois modelos de visualizações desenvolvidos nesse projeto.



**Figura 26.** Visualização de dados em nuvem de bolhas. A imagem ilustra a quantidade que cada tipo de item possui. Doujinshi são mangás distribuídos independentemente, geralmente produzidos por artistas iniciantes.



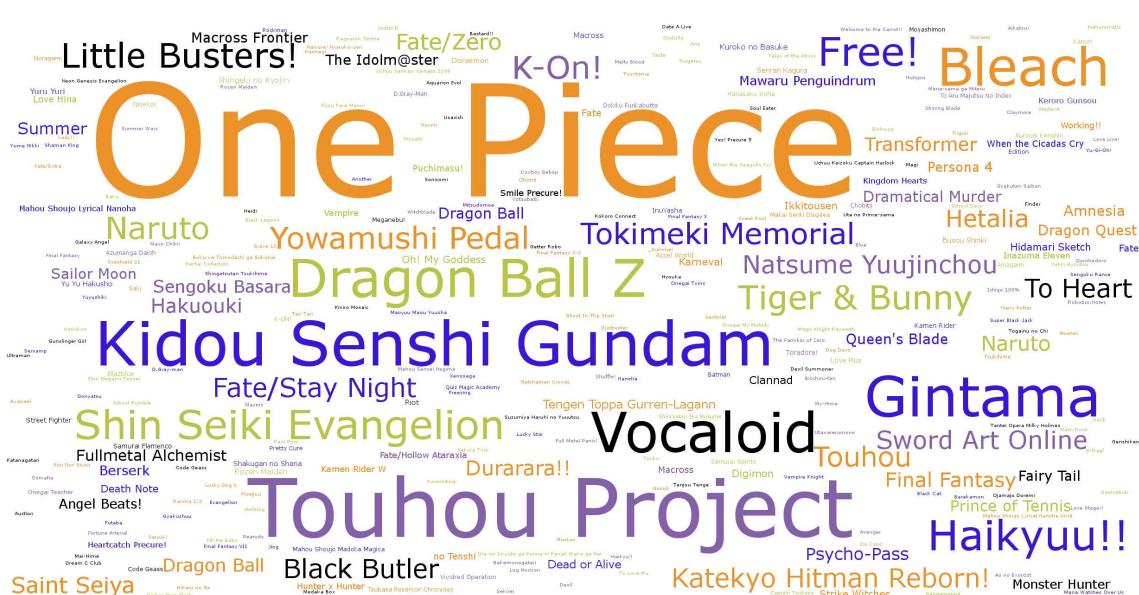
**Figura 27.** Visualização de dados em nuvem de palavras. A imagem ilustra a quantidade que cada tipo de item possui.



**Figura 28.** Visualização de dados em nuvem de bolhas. A imagem ilustra a quantidade de profissionais envolvidos na produção de itens.



**Figura 29.** Visualização de dados em nuvem de palavras. A imagem ilustra a quantidade de profissionais envolvidos na produção de itens.



**Figura 30.** Visualização de dados em nuvem de palavras. A imagem ilustra a quantidade de itens em cada franquia.



**Figura 31.** Visualização de dados em nuvem de bolhas. A imagem ilustra a quantidade de itens em cada franquia.



**Figura 32.** Visualização de dados em nuvem de palavras com máscara. A imagem ilustra a quantidade de itens em cada franquia.

Para ambos os modelos o tempo de processamento é similar, e dependendo da quantidade de objetos (palavras ou círculos), a serem utilizados, podem ser processados em alguns minutos, em casos com 10 objetos, ou em 24 horas, em casos com 3.000 objetos. A maior parte do tempo é consumida pela busca por espaço disponível.

## Referências

- [1] P. Viola; M. Jones. *Rapid object detection using a boosted cascade of simple features*. In IEEE Computer Vision and Pattern Recognition (pp. I:511–518), 2001.
- [2] Konstantinos G. Derpanis. *Integral image-based representations*, Department of Computer Science and Engineering, York University, New York, 2007.
- [3] Marco Lui; Timothy Baldwin. *langid.py: An Off-the-shelf Language Identification Tool*, Department of Computing and Information Systems, University of Melbourne, VIC 3010, Australia.
- [4] Scrapy developers. *Scrapy Tutorial*, 2014, disponível em <http://doc.scrapy.org/en/latest/intro/tutorial.html>.
- [5] Fredrik Lundh. *The Python Imaging Library Handbook*, 2014, disponível em <http://effbot.org/imagingbook/>.