***Regras de mapeamento***

**Mapeamento de Entidades**

Com base na modelagem conceitual e nas especificações no dicionário de dados decidimos optar pelas seguintes regras.

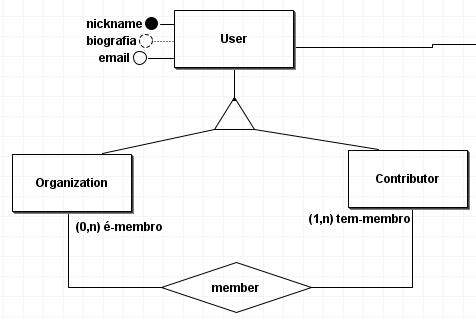
Para melhor apresentação das regras seguiremos a seguinte sequência de transformação do modelo ER em um modelo relacional:

* Entidades e respectivos atributos;
* Generalizações/especializações;
* Relacionamentos e respectivos atributos;

Notações:

* **Chave primário em negrito;**
* *Chave estrangeira em itálico;*
* Atributo opcional destacado com \*;
* Atributo único destacado com sublinhado;

**Entidade User: especialização**



Considerando a especialização que a entidade User possui, as distintas relações que Organization e Contributor participam e pelas operações que serão realizadas nessas tabelas decidimos transformar a especialização nas seguintes tabelas:

(para melhor as relações das tabelas, criamos um atributo ID. Será usado no lugar do nickname, que se transformará em chave alternativa)

Users (**id**, nickname, biografia, email)

Contributors (**id**, *idUser*)

references Users (idUsers);

Organizations (**id**, *idUser*)

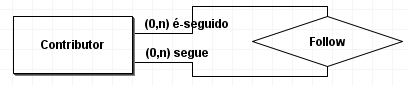
references Users (idUsers);

Members (***idContrib***, ***idOrg***)

references Contributor(idContrib),

references Organization(idOrg);

**Auto Relacionamento Follow:**



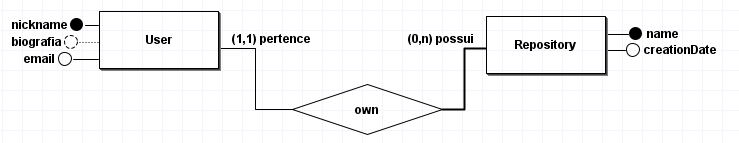
Criado uma tabela para o relacionamento por se tratar de um Relacionamento N-M:

Follows (***idFollower***, ***idFollowed***, date)

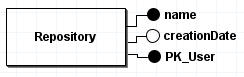
references Contributors(IdFollower),

references Contributors(idFollowed);

**Entidade Repository**



Mapeado para:



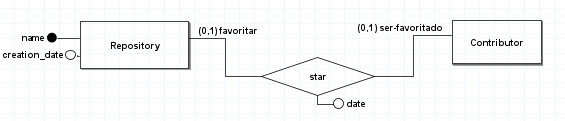
Esse mapeamento foi feito considerando a resolução do relacionamento 1-n e a Entidade Fraca. Dessa forma, cada repositório será único dentre os repositórios do usuário.

Repositories (**idRepository**, *idUser*, name, creationDate)

*references Users(idUser)*

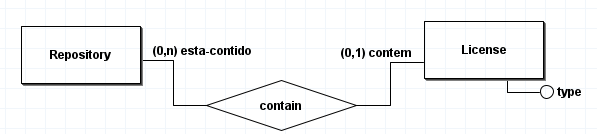
Escolhemos essa resolução para tabela Repository pois ela participará de muitos relacionamentos posteriormente (Dar exemplos).

**Relação entre Contributor e Repository**



Mapeado para:

**Entidade License e relação com Repository**



Mapeado para:

License (**id**, type)

Repositories (**idRepository**, *idUser*, name, creationDate, *idLicense*)

*references Users (idUser),*

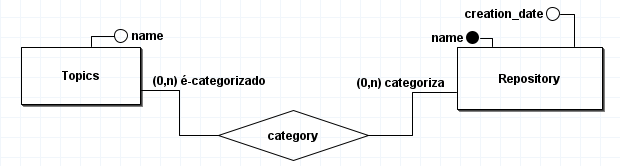
*references Licence (idLicense);*

Alterações feitas:

* Repository recebeu novo atributo: IdLicense

Decidimos por esse mapeamento pois o Relacionamento é 1-N, portanto, representando assim, o desejado.

**Entidade Topics e relação com Repository**



Criado uma tabela para o relacionamento por se tratar de um Relacionamento N-M:

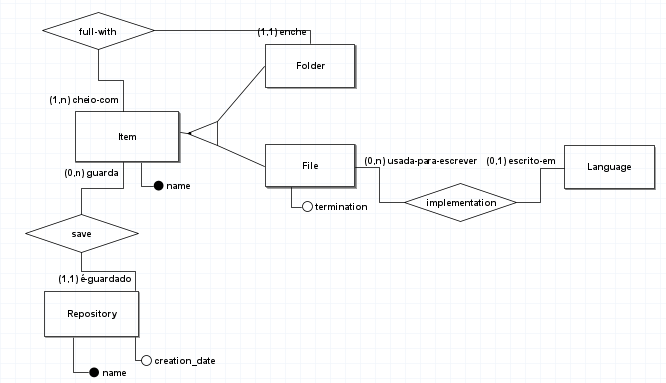
Topics (**cod**, name)

Category (***idRepository***, ***codTopic***)

*references Repository (idRepository),*

*references Topics (codTopic);*

**Estruturação do repositório**



**Entidade Item: especialização e relação com Repository**

Mapeado para:

Items (**id**, name, *idRepository*)

*references Repository (idRepository);*

Folders (**idFolder**, *idItem*)

*references Items (idItem);*

Files (**id**, *idItem*, termination)

*references Item (idItem);*

Por se tratar de um Relacionamento 1-N (Obrigatório do “lado 1” / Opcional do “lado N”), mapeamos dessa maneira. Optamos por essa alternativa considerando o relacionamento de Item com Repository. Além disso, consultar quantos arquivos um repositório tem ou quantas pastas ele possui fica trivial.

**Entidade Language:**

Mapeado para:

Languages (**cod**, name)

Implementations (***idFile,*** *codLanguage*)

*references Files (idFile),*

*references Languages (codLanguage);*

Utilizamos esse mapeamento pensando no futuro: posteriormente será implementado uma consulta importante com o objetivo de obter o percentual das linguagens utilizadas em um repositório. A relação entre Files e Languages modelado dessa forma, visa aumentar o desempenho dessa consulta em específico.

**Entidade Issue e relações**

