



INSTITUTO FEDERAL DA PARAÍBA - IFPB
Unidade Acadêmica de Informação e Comunicação
CST em Sistemas para Internet

Disciplina: Banco de Dados II

Professores: Damires e Thiago

Grupo: Felipe de Brito Lira Cavalcante e Thayná Lopes Tolentino

Roteiro para Miniprojeto de BD baseado em Documentos (10,0)

** Entregar textos, diagramas e códigos através de um documento Google Docs. Os scripts para rodar devem ser também entregue em documento .js.

1. Modelo Entidade-Relacionamento em nível Conceitual (diagrama) atualizado para mini projeto, conforme regras de negócio do projeto de Banco de Dados Relacional (1,0)

a. Obrigatório ter:

- i. Duas a três entidades
- ii. Vários atributos simples
- iii. Pelo menos um atributo multivalorado
- iv. Pelo menos um atributo composto
- v. Pelo menos um atributo opcional
- vi. Pelo menos um relacionamento 1:N com sentido de agregação ou composição (possui, tem, contém...)

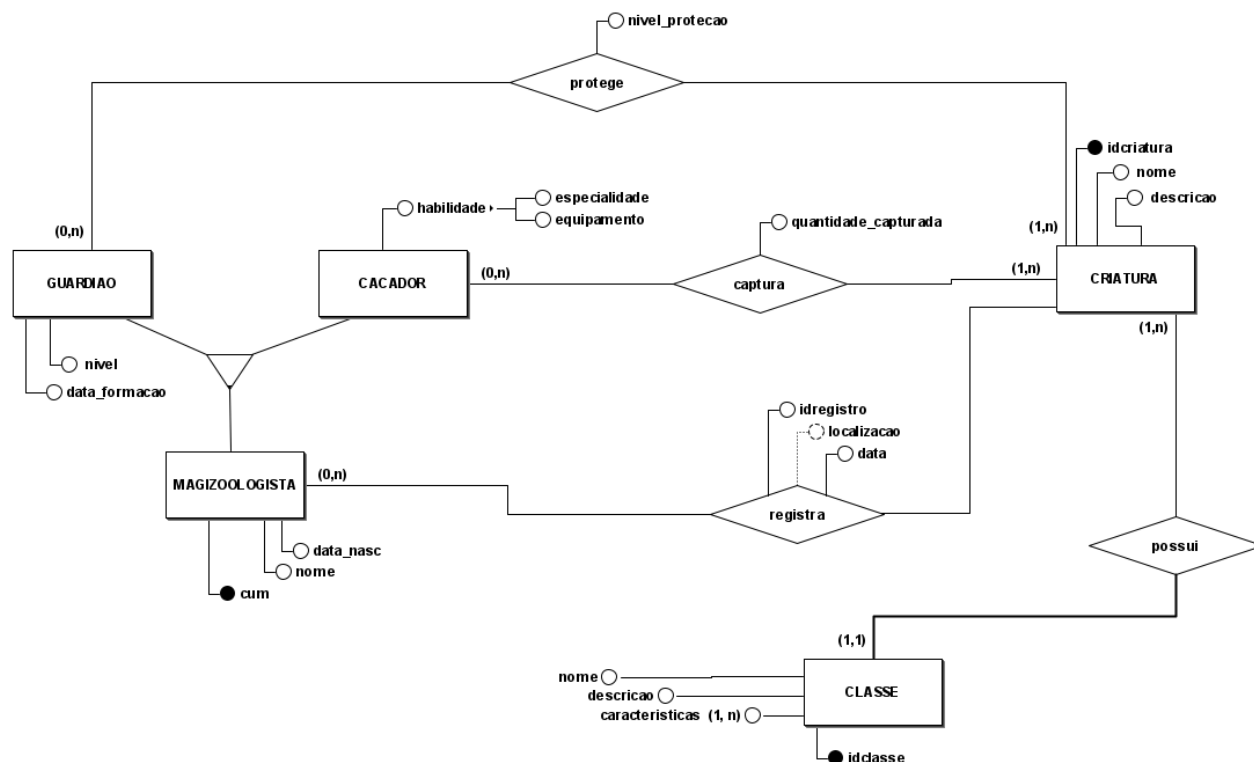
b. Opcional:

- i. Hierarquia de generalização/especialização
- ii. Relacionamentos N:N, N:1, 1:1

Obs:

- *As entidades e relacionamentos devem ser especificadas conforme regras de negócio e descrição realizada no projeto de BDR (Template1);*
- *Escolher duas a três entidades do projeto de BDR que atendam ao que se pede.*
- *Lembrar que o modelo ER em nível conceitual **INDEPENDENTE de tecnologia**, por isso podemos usá-lo tanto para projeto de bancos de dados relacionais como para qualquer outro modelo de representação/persistência.*

Novo diagrama conceitual:



2. Mapeamento entre as entidades e relacionamentos do modelo conceitual e coleções de documentos no MongoDB.

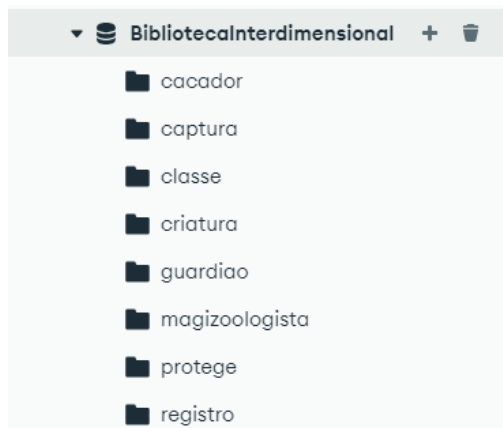
| Modelo Conceitual | Tipo no MC | MongoDB | Tipo no MongoDB | Observação |
|---------------------------|------------------------|-----------------|-----------------|--|
| Criatura | Entidade | Criatura | Coleção | Principal entidade do projeto |
| Classe | Entidade Fraca | Classe | Coleção | Entidade vira campo id obrigatório em criatura |
| caracteristicas | Atributo multivalorado | Campo de Classe | Campo – Array | Campo obrigatório |
| Magizoologista | Entidade | Magizoologista | Coleção | Entidade Generalizada |
| Registra [Magizoologista] | Relacionamento | Registro | Coleção | Tem as chaves primárias das |

| | | | | |
|------------------------|------------------------|-------------------|-------------------------------|--|
| | | | | tabelas relacionadas |
| localizacao [registra] | Atributo opcional | Campo de registro | Campo simples opcional | Campo opcional |
| Cacador | Entidade especializada | Cacador | Coleção | Entidade Especializada que tem atributo composto |
| habilidade | Atributo composto | Campo de cacador | Campo – estrutura (embutido) | Campo composto por especialidade e equipamento |
| captura [cacador] | relacionamento | Captura | Coleção | Possui as PKs das entidades relacionadas. |
| guardião | Entidade especializada | guardiao | Coleção | Entidade Especializada |
| protege [guardião] | relacionamento | protege | Coleção | Possui as PKs das entidades relacionadas. |

3. Implementação do projeto no MongoDB

a. Objetos básicos:

- i. Coleção(ões) de documentos: criar coleção(ões) conforme mapeamentos realizados e decisões de projeto (1,0)



- ii. 2 índices com apresentação de justificativa (1,0).

```
BibliotecaInterdimensional> db.criatura.createIndex({ "nome": 1 });
```

- Índice em **nome** na coleção **Criatura**: Melhora a performance e eficiência das buscas na coleção criatura, permitindo maior agilidade na busca a partir do nome dela ao invés de seu id.

```
BibliotecaInterdimensional > db.magizoologista.createIndex({ "nome": 1 });
```

- Índice em **nome** na coleção **Magizoologista**: Melhora a performance e eficiência das buscas na coleção magizoologista, permitindo agilidade nas buscas de um magizoologista pelo seu nome, já que seus cums são extensos demais e os magizoologistas são frequentemente procurados através do seu nome.

b. Operações de manipulação (CRUD) (6,0):

*** Todas as operações devem apresentar seu enunciado e sua solução*

*** Os comandos (todos) devem fazer sentido à aplicação e seus requisitos*

- i. Pelo menos 5 comandos de inserções para cada coleção (0,6).
- ii. Consultas diversas de acordo com o que se pede (5,4):
 - 2 consultas com pelo menos filtros diversos (IN, GT, etc), sem projeção;
 - 2 consultas com pelo menos filtros diversos e com projeção;
 - 1 consulta com apenas projeção (sem filtro);
 - 1 consulta com pelo menos acesso a elemento de array;
 - 1 consulta com pelo menos acesso a estrutura/objeto embutido;
 - 1 consulta com pelo menos sort e limit e filtros e projeções;
 - 1 consulta com pelo menos aggregate e lookup;
 - 1 outra consulta a seu critério, dentro do contexto da aplicação.

Consultas estão no arquivo Script.js, com suas devidas explicações.