



**UNIVERSIDADE FEDERAL DE OURO PRETO**  
**INSTITUTO DE CIÊNCIAS EXATAS E BIOLÓGICAS**  
**DEPARTAMENTO DE COMPUTAÇÃO**

Halliday Gauss Costa dos Santos.  
Marcos Geraldo Braga Emiliano  
Matheus Henrique dos Santos Almeida  
Lucas Urzedo da Silva Paulino  
Vivyann Fernandes Cedraz

**TRABALHO PRÁTICO I**

**Ouro Preto**  
**2021**

## **SUMÁRIO**

- I. INTRODUÇÃO**
- II. MINI-MUNDO**
- III. MODELO CONCEITUAL**
- IV. DICIONÁRIO DE DADOS**
  - TABELA DE ENTIDADES**
  - RELACIONAMENTOS COM ATRIBUTOS**
  - RELACIONAMENTOS SEM ATRIBUTOS**
- V. MODELO RELACIONAL**
- VI. ESQUEMA FÍSICO**
- VII. REFERÊNCIAS**

## **I. INTRODUÇÃO**

O trabalho consiste no planejamento e elaboração dos requisitos de dados de um problema apresentado no Mini Mundo e do consequente esquema conceitual Entidade Relacionamento Estendido (ERE) do futuro Banco de Dados que será gerado para a aplicação. Em linhas gerais o problema se baseia em uma WIKI Fandom, onde pessoas podem visualizar informações e comentar sobre Quadrinhos, Filmes, Séries, Personagens e Cenários de diversas obras. Foi desenvolvido e apresentado o modelo entidade relacionamento estendido e o dicionário de dados referente ao Mini Mundo.

## **II. MINI-MUNDO**

A empresa “VASP” teve uma ideia de criação de um software web que gerencia uma Fandom de Séries, Filmes e Quadrinhos. A plataforma tem o intuito de exibir e contribuir com novas informações para os fãs de entretenimento. Nessa plataforma existem Comunidades, e cada comunidade é composta por informações sobre uma das seguintes formas de entretenimento:

- Séries: cada série possui um número de episódios, a avaliação, a sinopse, um nome e os gêneros que se encaixa, além do nome e duração de cada episódio.
- Filmes: cada filme possui um número de episódios, a avaliação, a sinopse, um nome, os gêneros que se encaixa, bilheteria e duração.
- Personagens: cada personagem possui um número de identificação, a sua habilidade, o seu nome, a sua história e sua idade.
- Quadrinhos: cada quadrinho possui uma editora, um nome, um volume, um autor (com seu nome, nacionalidade e avaliação) e uma identificação.
- Cenários: cada cenário possui uma descrição, origem, nome e uma identificação.

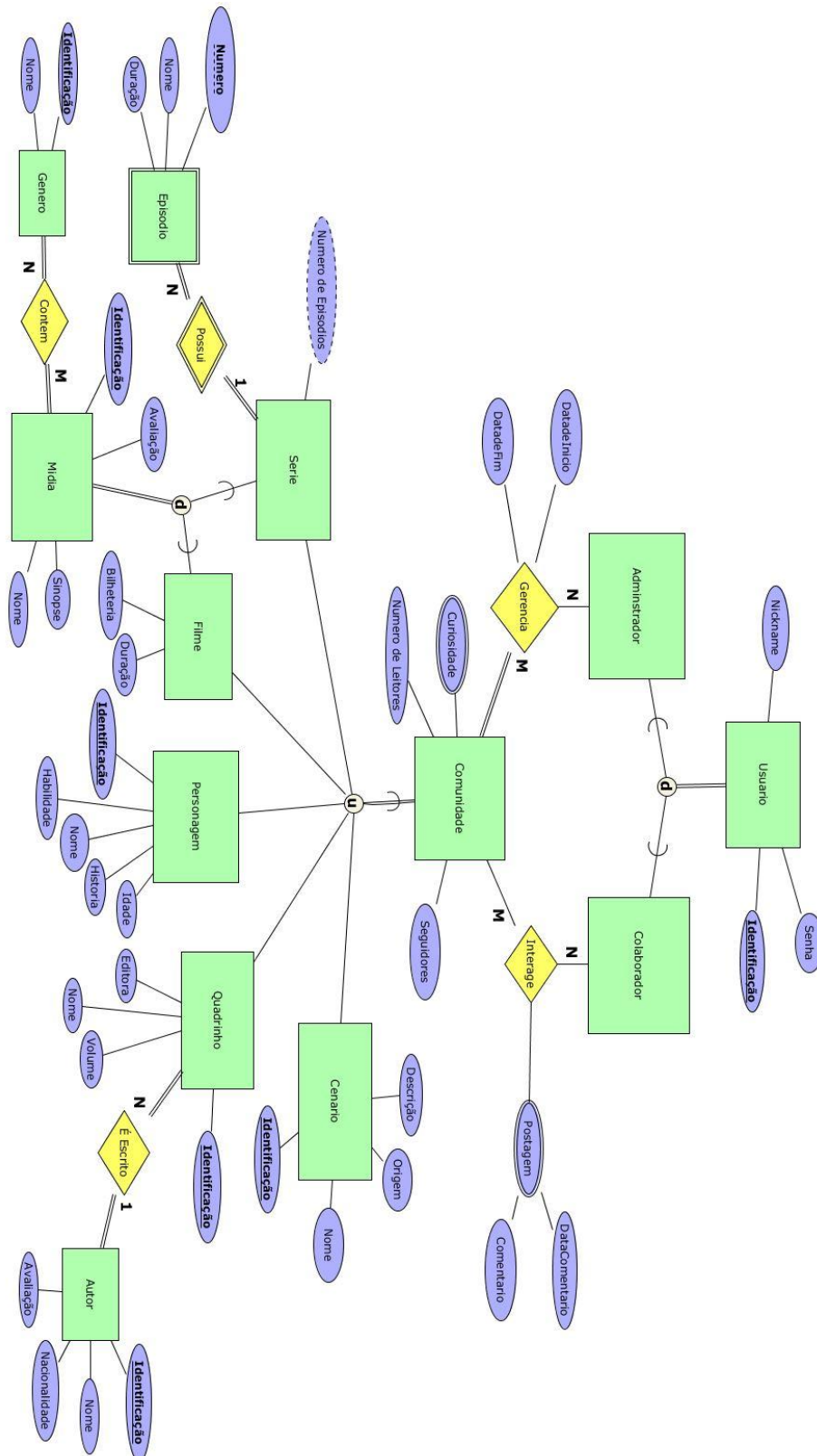
A WIKI deve mostrar as curiosidades, o número de leitores e seguidores de cada comunidade. O Sistema da WIKI Fandom deve ter uma forma de verificação para

que somente usuários administradores possam gerenciar as comunidades, devendo mostrar a data de início e fim da gestão de cada comunidade entre os administradores. O Sistema deve aceitar também o login de colaboradores, e somente esse tipo de usuário pode publicar nos tópicos, ou seja, interagir nas comunidades por meio de comentários, e cada comentário deve ter uma data em que foi publicado. Todo usuário deve possuir um nickname, uma senha e uma identificação. Um simples leitor não precisa de login, podendo navegar no sistema e fazer a leitura dos dados da comunidade que mais lhe interessa.

Dado o problema, foi solicitado pela empresa um Banco de Dados que atenda os requisitos supracitados para que futuramente a aplicação possa ser implementada.

### III. MODELO CONCEITUAL

A figura abaixo apresenta o esquema conceitual ERE do Mini Mundo proposto:



## IV. DICIONÁRIO DE DADOS

### 1. TABELA DE ENTIDADES

#### 1.1 Usuário

O tipo de entidade Usuário representa um usuário do sistema e seus campos de atributos.

Atributo	Tipo de dado	Comprimento	Restrições	Tipo de atributo	Descrição
Nickname	Alpha numérico	10	NOT NULL, ÚNICO	Simples, mono-valorado, armazenado	Nome único atribuído ao usuário
Senha	Alpha numérico	8	NOT NULL	Simples, mono-valorado, armazenado	Senha de acesso ao sistema
Identificação	Inteiro	100	NOT NULL, ÚNICO, PK	Simples, mono-valorado, armazenado	Identificador único de cada usuário

##### 1.1.1 Administrador

Subclasse de Usuário que representa um administrador.

##### 1.1.2 Colaborador

Subclasse de Usuário que representa um colaborador.

#### 1.2 Comunidade

Tipo de união total Comunidade de Cenário, Quadrinho, Personagem, Série e Filme.

<b>Atributo</b>	<b>Tipo de dado</b>	<b>Comprimento</b>	<b>Restrições</b>	<b>Tipo de atributo</b>	<b>Descrição</b>
Curiosidade	Alpha numérico	240	NULL	Simples, multi-valorado, armazenado	Curiosidades escritas por colaboradores
Seguidores	Inteiro	20	NOT NULL	Simples, mono-valorado, armazenado	Número de seguidores de uma comunidade
Número de leitores	Inteiro	20	NOT NULL	Simples, mono-valorado, armazenado	Número de leitores de uma comunidade

### 1.2.1 Cenário

Tipo de entidade que representa um cenário no sistema, sendo uma superclasse da união Comunidade.

<b>Atributo</b>	<b>Tipo de dado</b>	<b>Comprimento</b>	<b>Restrições</b>	<b>Tipo de atributo</b>	<b>Descrição</b>
Descrição	Alpha numérico	240	NOT NULL	Simples, mono-valorado, armazenado	Descrição dos elementos que compõe o cenário
Origem	Alpha numérico	100	NOT NULL	Simples, mono-valorado, armazenado	Local de origem do cenário
Nome	Alpha numérico	50	NOT NULL	Simples, mono-valorado, armazenado	Nome do cenário
Identificação	Inteiro	20	NOT NULL, ÚNICO, PK	Simples, mono-valorado, armazenado	Identificador único de cada cenário

### 1.2.2 Quadrinho

Tipo de entidade que representa um quadrinho no sistema, sendo uma superclasse da união Comunidade.

Atributo	Tipo de dado	Comprimento	Restrições	Tipo de atributo	Descrição
Editora	Alpha numérico	50	NOT NULL	Simples, mono-valorado, armazenado	Nome da editora que publicou o quadrinho
Nome	Alpha numérico	100	NOT NULL	Simples, mono-valorado, armazenado	Nome do quadrinho
Volume	Inteiro	5	NOT NULL	Simples, mono-valorado, armazenado	Número do volume
Identificação	Inteiro	20	NOT NULL, ÚNICO, PK	Simples, mono-valorado, armazenado	Identificador único de cada quadrinho

#### 1.2.2.1 Autor

Entidade que representa o autor de um quadrinho.

Atributo	Tipo de dado	Comprimento	Restrições	Tipo de atributo	Descrição
Avaliação	Float	5	NOT NULL	Simples, mono-valorado, armazenado	Nota geral do autor
Nacionalidade	Alpha numérico	50	NOT NULL	Simples, mono-valorado, armazenado	Nacionalidade do autor
Nome	Alpha numérico	50	NOT NULL	Simples, mono-valorado, armazenado	Nome do Autor
Identificação	Inteiro	20	NOT NULL, ÚNICO, PK	Simples, mono-valorado, armazenado	Identificador único de cada autor



### 1.2.3 Personagem

Tipo de entidade que representa um personagem no sistema, sendo uma superclasse da união Comunidade.

Atributo	Tipo de dado	Comprimento	Restrições	Tipo de atributo	Descrição
História	Alpha numérico	500	NOT NULL	Simples, mono-valorado, armazenado	Biografia do personagem
Nome	Alpha numérico	50	NOT NULL	Simples, mono-valorado, armazenado	Nome do personagem
Habilidade	Alpha numérico	100	NOT NULL	Simples, mono-valorado, armazenado	Características próprias do personagem
Idade	Inteiro	20	NOT NULL	Simples, mono-valorado, armazenado	Idade do personagem
Identificação	Inteiro	20	NOT NULL, ÚNICO, PK	Simples, mono-valorado, armazenado	Identificador único de cada personagem

### 1.2.4 Mídia

Tipo de entidade que representa uma mídia no sistema.

Atributo	Tipo de dado	Comprimento	Restrições	Tipo de atributo	Descrição
Avaliação	Float	3	NOT NULL	Simples, mono-valorado, armazenado	Pontuação atribuída a mídia
Sinópsse	Alpha numérico	240	NOT NULL	Simples, mono-valorado, armazenado	Resumo da mídia
Nome	Alpha numérico	50	NOT NULL	Simples, mono-valorado, armazenado	Nome da mídia

Identificação	Inteiro	20	NOT NULL, ÚNICO, PK	Simples, mono-valorado, armazenado	Identificador único de cada mídia
---------------	---------	----	------------------------	--	---

#### 1.2.4.1 Gênero

Entidade que representa o tipo de gênero de uma mídia.

Atributo	Tipo de dado	Comprimento	Restrições	Tipo de atributo	Descrição
Nome	Alpha numérico	50	NOT NULL	Simples, mono-valorado, armazenado	Nome do gênero
Identificação	Inteiro	20	NOT NULL, ÚNICO, PK	Simples, mono-valorado, armazenado	Identificador único de cada gênero

#### 1.2.4.2 Série

Subclasse de mídia que representa uma série no sistema, sendo uma superclasse da união Comunidade.

Atributo	Tipo de dado	Comprimento	Restrições	Tipo de atributo	Descrição
Número de episódios	Inteiro	5	NOT NULL	Simples, mono-valorado, derivado	Quantidade de episódios que possui uma série

##### 1.2.4.2.1 Episódio

Entidade fraca que representa um episódio de uma série.

<b>Atributo</b>	<b>Tipo de dado</b>	<b>Comprimento</b>	<b>Restrições</b>	<b>Tipo de atributo</b>	<b>Descrição</b>
Nome	Alpha numérico	100	NOT NULL	Simples, mono-valorado, armazenado	Nome do episódio
Número	Inteiro	5	NOT NULL, ID	Simples, mono-valorado, armazenado	Número do episódio
Duração	Hora	hh:mm:ss	NOT NULL	Simples, mono-valorado, armazenado	Tempo de duração de um episódio

### 1.2.4.3 Filme

Subclasse de mídia que representa um filme no sistema, sendo uma superclasse da união Comunidade.

<b>Atributo</b>	<b>Tipo de dado</b>	<b>Comprimento</b>	<b>Restrições</b>	<b>Tipo de atributo</b>	<b>Descrição</b>
Bilheteria	Inteiro	10	NOT NULL	Simples, mono-valorado, armazenado	Número que pessoas que assistiram o filme na semana do lançamento
Duração	Hora	hh:mm:ss	NOT NULL	Simples, mono-valorado, armazenado	Tempo de duração do filme

## 2. RELACIONAMENTOS COM ATRIBUTOS

### 2.1 Relacionamento “Interage”, entre entidades Colaborador e Comunidade

Relacionamento com restrição de cardinalidade N:M e de participação parcial-parcial.

Atributo	Tipo de dado	Comprimento	Restrições	Tipo de atributo	Descrição
Postagem (Comentario, DataComentario)	Alpha numérico, Data	500, aaa:mm:dd	NOT NULL, NOT NULL	Composto, multi-valorado, armazenado	Postagem sobre uma comunidade

## 2.1 Relacionamento “Gerencia”, entre entidades Administrador e Comunidade

Relacionamento com restrição de cardinalidade N:M e de participação parcial-total.

Atributo	Tipo de dado	Comprimento	Restrições	Tipo de atributo	Descrição
DataInicio	Data	aaa:mm:dd	NOT NULL	Simples, mono-valorado, armazenado	Data de início do gerenciamento
DataFim	Data	aaa:mm:dd	NULL	Simples, mono-valorado, armazenado	Data de fim do gerenciamento

## 3. RELACIONAMENTOS SEM ATRIBUTOS

### 3.1 Relacionamento identificador “Possui”, entre a entidade forte Série e a entidade fraca Episódio.

Relacionamento identificador com restrição de cardinalidade 1:N e de participação total-total.

<b>Relacionamento</b>	<b>Entidade 1</b>	<b>Entidade 2</b>	<b>Restrição de cardinalidade</b>	<b>Restrição de participação</b>	<b>Descrição</b>
Possui	Série	Episódio	1:N	Total-Total	Relacionamento identificador entre Série e Episódio.

### **3.2 Relacionamento “Contém”, entre as entidades Mídia e Gênero.**

Relacionamento com restrição de cardinalidade N:M e de participação total-total.

<b>Relacionamento</b>	<b>Entidade 1</b>	<b>Entidade 2</b>	<b>Restrição de cardinalidade</b>	<b>Restrição de participação</b>	<b>Descrição</b>
Contém	Mídia	Gênero	N:M	Total-Total	Relacionamento entre Mídia e Gênero.

### **3.3 Relacionamento “É Escrito”, entre as entidades Quadrinho e Autor.**

Relacionamento com restrição de cardinalidade N:1 e de participação total-total.

<b>Relacionamento</b>	<b>Entidade 1</b>	<b>Entidade 2</b>	<b>Restrição de cardinalidade</b>	<b>Restrição de participação</b>	<b>Descrição</b>
É Escrito	Quadrinho	Autor	N:1	Total-Total	Relacionamento entre Quadrinho e Autor.

## V. MODELO RELACIONAL

Dado o Modelo Entidade Relacionamento Estendido é possível, através do Projeto Lógico, criar o Modelo Relacional utilizando as regras de mapeamento. Dado o Modelo ERE acima, o Modelo Relacional ficará da seguinte maneira:

Usuario(Identificação, Senha, Nickname)

Administrador(Identificação)

Administrador[Identificação]<sup>p</sup> → Usuario[Identificação]

Colaborador(Identificação)

Colaborador[Identificação]<sup>p</sup> → Usuario[Identificação]

Comunidade(Identificação, Seguidores, NumerodeLeitores)

Curiosidade(Identificação, Curiosidade)

Curiosidade [Identificação]<sup>p</sup> → Comunidade[Identificação]

Interage( IdentificaçãoCola, IdentificaçãoComuni)

Interage [IdentificaçãoCola]<sup>p</sup> → Usuario [Identificação]

Interage [IdentificaçãoComuni]<sup>p</sup> → Comunidade[Identificação]

Postagem(IdentificaçãoInterCola, IdentificaçãoInterComuni, IdentificaçãoPost,  
DataComentario, Comentario)

Postagem [IdentificaçãoInterCola]<sup>p</sup> → Usuario [IdentificaçãoCola]

Postagem [IdentificaçãoInterComuni]<sup>p</sup> → Comunidade [IdentificaçãoComuni]

Gerencia(IdentificaçãoAdm, IdentificaçãoComuni,DataInicio,DataFim)

Gerencia [IdentificaçãoAdm]<sup>p</sup> → Usuario [Identificação]

Gerencia [IdentificaçãoComuni]<sup>p</sup> → Comunidade [Identificação]

Midia(Identificação,Avaliacao,Sinopse,Nome)

Midia [Identificação]<sup>p</sup> → Comunidade [Identificação]

Genero(Identificação,Nome)

Contem(IdentificaçãoMidia, IdentificaçãoGenero)

Contem [IdentificaçãoMidia]<sup>p</sup> → Midia[Identificação]

Contem [IdentificaçãoGenero]<sup>p</sup> → Genero[Identificação]

Filme(Identificação, Bilheteria,Duração)

Filme [Identificação]<sup>p</sup> → Midia[Identificação]

Serie(Identificação)

Serie [Identificação]<sup>p</sup> → Midia[Identificação]

Episodio(IdentificaçãoSerie, Numero, Nome, Duração)

Episodio [IdentificaçãoSerie]<sup>p</sup> → Serie [Identificação]

Personagem(Identificação, Nome, Idade, Historia, Habilidade)

Personagem [Identificação]<sup>p</sup> → Comunidade[Identificação]

Autor(Identificação, Nome, Nacionalidade, Avaliação)

Quadrinho(Identificação, IDAutor, Editora, Nome, Volume)

Quadrinho [Identificação]<sup>p</sup> → Comunidade[Identificação]

Quadrinho [IDAutor]<sup>B</sup> → Autor[Identificação]

Cenario(Identificação, Nome, Origem, Descricao)

Cenario [Identificação]<sup>p</sup> → Comunidade[Identificação]



## VI. ESQUEMA FÍSICO

Dado o Modelo Relacional foi criado, através do Projeto Físico, o Esquema Físico do Banco de Dados na linguagem SQL. Dado o Modelo Relacional acima, o Esquema Físico em SQL terá a seguinte estrutura:

```
CREATE TABLE Usuario (  
    Nickname          VARCHAR(10)      NOT NULL,  
    Senha              VARCHAR(8)       NOT NULL,  
    Identificacao      SERIAL,  
    CONSTRAINT PK_Usuario PRIMARY KEY(Identificacao),  
    CONSTRAINT UK_Nickname UNIQUE (Nickname)  
);  
  
CREATE TABLE Administrador (  
    Identificacao      SERIAL,  
    CONSTRAINT PK_Administrador PRIMARY KEY(Identificacao),  
    CONSTRAINT FK_Usuario FOREIGN KEY(Identificacao)  
        REFERENCES Usuario (Identificacao)  
        ON DELETE CASCADE  
        ON UPDATE CASCADE,  
    CONSTRAINT UK_Adm UNIQUE (Identificacao)  
);  
  
CREATE TABLE Colaborador (  
    Identificacao      SERIAL,  
    CONSTRAINT PK_Colaborador PRIMARY KEY(Identificacao),  
    CONSTRAINT FK_Usuario FOREIGN KEY(Identificacao)  
        REFERENCES Usuario (Identificacao)
```

```

        ON DELETE CASCADE

        ON UPDATE CASCADE,

        CONSTRAINT UK_Col UNIQUE(Identificacao)
);

CREATE TABLE Comunidade (

    Identificacao SERIAL,

    Seguidores INT NOT NULL,

    Numero_de_leitores INT NOT NULL,

    CONSTRAINT PK_Comunidade PRIMARY KEY(Identificacao)
);

CREATE TABLE Curiosidade (

    Identificacao SERIAL,

    Curiosidade VARCHAR(240) NULL,

    CONSTRAINT FK_Comunidade FOREIGN KEY(Identificacao)

        REFERENCES Comunidade (Identificacao)

        ON DELETE CASCADE

        ON UPDATE CASCADE,

        CONSTRAINT PK_Curiosidade PRIMARY
KEY(Curiosidade,Identificacao)
);

CREATE TABLE Interage (

    IdentificacaoColaborador SERIAL,

    IdentificacaoComunidade SERIAL,

    CONSTRAINT FK_Comunidade FOREIGN KEY(IdentificacaoComunidade)

        REFERENCES Comunidade (Identificacao)

```

```

        ON DELETE CASCADE

        ON UPDATE CASCADE,

                                CONSTRAINT        FK_Colaborador        FOREIGN
KEY(IdentificacaoColaborador)

        REFERENCES Colaborador (Identificacao)

        ON DELETE CASCADE

        ON UPDATE CASCADE,

                                CONSTRAINT        PK_Interage        PRIMARY
KEY(IdentificacaoComunidade,IdentificacaoColaborador)
);

CREATE TABLE Postagem (

        IdentificacaoInterageColaborador SERIAL,

        IdentificacaoInterageComunidade SERIAL,

        IdentificacaoPostagem SERIAL,

        DataComentario DATE NOT NULL,

        Comentario VARCHAR(500) NOT NULL,

                                CONSTRAINT        FK_Interage        FOREIGN
KEY(IdentificacaoInterageColaborador,
IdentificacaoInterageComunidade)

        REFERENCES Interage (IdentificacaoColaborador,
IdentificacaoComunidade)

        ON DELETE CASCADE

        ON UPDATE CASCADE,

                                CONSTRAINT        PK_Postagem        PRIMARY
KEY(IdentificacaoInterageComunidade,IdentificacaoInterageColaborad
or,IdentificacaoPostagem)
);

CREATE TABLE Gerencia (

```

```

IdentificacaoAdministrador SERIAL,

IdentificacaoComunidade SERIAL,

DataInicio DATE NOT NULL,

DataFim DATE NULL,

        CONSTRAINT    FK_IdentificacaoAdminstrador    FOREIGN
KEY(IdentificacaoAdministrador)

        REFERENCES Administrador (Identificacao)

        ON DELETE CASCADE

        ON UPDATE CASCADE,

        CONSTRAINT    FK_IdentificacaoComunidade    FOREIGN
KEY(IdentificacaoComunidade)

        REFERENCES Comunidade (Identificacao)

        ON DELETE CASCADE

        ON UPDATE CASCADE,

        CONSTRAINT    PK_Gerencia    PRIMARY
KEY(IdentificacaoAdministrador,IdentificacaoComunidade)
);

CREATE TABLE Midia (

    Avaliacao    NUMERIC(3,1)    NOT NULL,

    Sinopse    VARCHAR(240)    NOT NULL,

    Nome    VARCHAR(50)    NOT NULL,

    Identificacao    SERIAL,

    CONSTRAINT FK_IdentificacaoMidia FOREIGN KEY(Identificacao)

        REFERENCES Comunidade (Identificacao)

        ON DELETE CASCADE

        ON UPDATE CASCADE,

    CONSTRAINT PK_Midia PRIMARY KEY(Identificacao)
);

```

```

CREATE TABLE Genero (
    Nome          VARCHAR(50)      NOT NULL,
    Identificacao SERIAL,
    CONSTRAINT PK_Genero PRIMARY KEY(Identificacao)
);

```

```

CREATE TABLE Contem (
    IdentificacaoMidia SERIAL,
    IdentificacaoGenero SERIAL,
    CONSTRAINT FK_IdentificacaoMidia FOREIGN
KEY(IdentificacaoMidia)
    REFERENCES Midia (Identificacao)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT FK_IdentificacaoGenero FOREIGN
KEY(IdentificacaoGenero)
    REFERENCES Genero (Identificacao)
    ON DELETE CASCADE
    ON UPDATE CASCADE,
    CONSTRAINT PK_Contem PRIMARY
KEY(IdentificacaoMidia,IdentificacaoGenero)
);

```

```

CREATE TABLE Filme (
    Bilheteria INT      NOT NULL,
    Duracao    INT      NOT NULL,
    Identificacao SERIAL,
    CONSTRAINT FK_IdentificacaoFilme FOREIGN KEY(Identificacao)

```

```

        REFERENCES Midia (Identificacao)

        ON DELETE CASCADE

        ON UPDATE CASCADE,

        CONSTRAINT PK_Filme PRIMARY KEY(Identificacao)
);

CREATE TABLE Serie (

    Identificacao    SERIAL,

    CONSTRAINT FK_IdentificacaoSerie FOREIGN KEY(Identificacao)

        REFERENCES Midia (Identificacao)

        ON DELETE CASCADE

        ON UPDATE CASCADE,

    CONSTRAINT PK_Serie PRIMARY KEY(Identificacao)
);

CREATE TABLE Episodio (

    Nome            VARCHAR(240)    NOT NULL,

    Numero          INT             NOT NULL,

    Duracao         TIME            NOT NULL,

    IdentificacaoSerie SERIAL,

        CONSTRAINT FK_IdentificacaoEpisodio FOREIGN
KEY(IdentificacaoSerie)

        REFERENCES Serie (Identificacao)

        ON DELETE CASCADE

        ON UPDATE CASCADE,

    CONSTRAINT PK_Episodio PRIMARY KEY(IdentificacaoSerie,Numero)
);

```

```

CREATE TABLE Personagem (

    Historia          VARCHAR(500)      NOT NULL,

    Nome              VARCHAR(50)        NOT NULL,

    Habilidade        VARCHAR(100)       NOT NULL,

    Idade             INT                 NOT NULL,

    Identificacao     SERIAL,

                                CONSTRAINT FK_IdentificacaoPersonagem FOREIGN
KEY(Identificacao)

    REFERENCES Comunidade (Identificacao)

    ON DELETE CASCADE

    ON UPDATE CASCADE,

    CONSTRAINT PK_Personagem PRIMARY KEY(Identificacao)

);

```

```

CREATE TABLE Autor (

    Avaliacao         NUMERIC(5,1)       NOT NULL,

    Nacionalidade     VARCHAR(50)         NOT NULL,

    Identificacao     SERIAL,

    Nome              VARCHAR(50)         NOT NULL,

    CONSTRAINT PK_Autor PRIMARY KEY(Identificacao)

);

```

```

CREATE TABLE Quadrinho (

    Editora           VARCHAR(50)         NOT NULL,

    Nome              VARCHAR(100)        NOT NULL,

    Volume            INT                 NOT NULL,

    Identificacao     SERIAL,

    IDAutor           SERIAL,

```

```

        CONSTRAINT FK_IdentificacaoQuadrinho FOREIGN
KEY(Identificacao)

        REFERENCES Comunidade (Identificacao)

        ON DELETE CASCADE

        ON UPDATE CASCADE,

CONSTRAINT FK_IDAutor FOREIGN KEY(IDAutor)

        REFERENCES Autor (Identificacao)

        ON UPDATE CASCADE,

CONSTRAINT PK_Quadrinho PRIMARY KEY(Identificacao)

);

CREATE TABLE Cenario (

        Descricao          VARCHAR(240)      NOT NULL,

        Origem              VARCHAR(100)      NOT NULL,

        Nome                VARCHAR(50)       NOT NULL,

        Identificacao       SERIAL,

CONSTRAINT FK_IdentificacaoCenario FOREIGN KEY(Identificacao)

        REFERENCES Comunidade (Identificacao)

        ON DELETE CASCADE

        ON UPDATE CASCADE,

CONSTRAINT PK_Cenario PRIMARY KEY(Identificacao)

);

```



## VII. REFERÊNCIAS

- 1- Tavares, Guilherme. Modelo de Entidades e Relacionamentos. Disponível em: [Microsoft PowerPoint - bd1\\_modelagem.ppt \[Modo de Compatibilidade\] \(ufop.br\)](#) . Acesso: 16 de outubro de 2021.
- 2- Tavares, Guilherme. Modelo de Entidades e Relacionamentos Estendidos. Disponível em: [Microsoft PowerPoint - bd1\\_modelagem.ppt \[Modo de Compatibilidade\] \(ufop.br\)](#) . Acesso em: 16 de outubro de 2021.
- 3- Tavares, Guilherme. SQL. Disponível em: [SQL \(Structured Query Language\) SQL \(Structured Query Language\) Esquema e Catálogo](#) . Acesso em: 01 de dezembro de 2021.
- 4- Tavares, Guilherme. Mapeamento ERE para Relacional. Disponível em: [Mapeamento ERE para Relacional](#) . Acesso em: 01 de dezembro de 2021.