

# Projeto de Bloco: Fundamentos de Dados [24E4\_5]

Prof. Luiz Paulo Maia



Elias Matos Martins

[Elias\\_Martins\\_PB\\_AT](#)



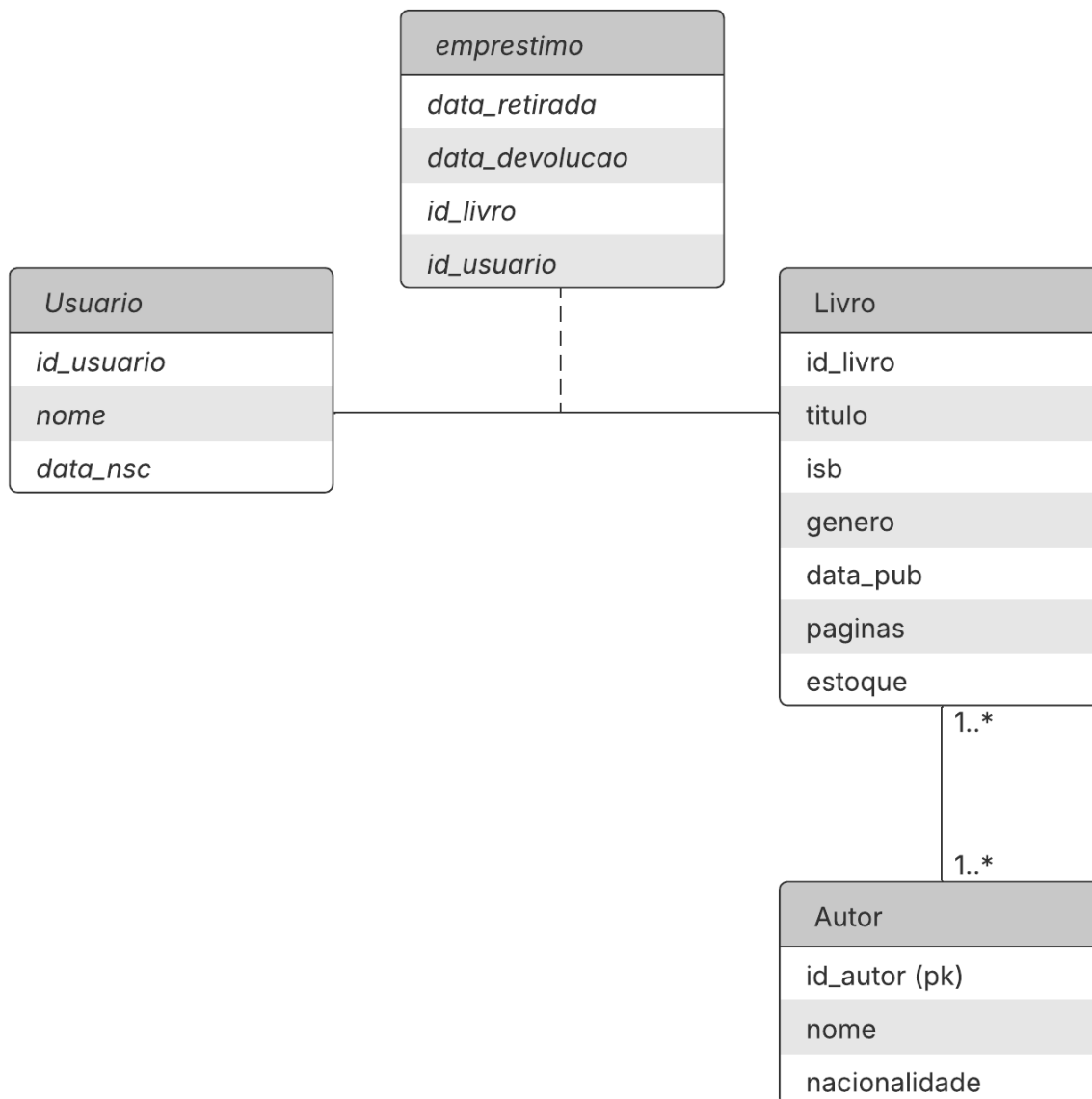
Instituto Infnet

7 de abril de 2025

# 1. Modelo Conceitual (DER - Diagrama Entidade Relacionamento)

## Biblioteca (Modelo Conceitual [DER])

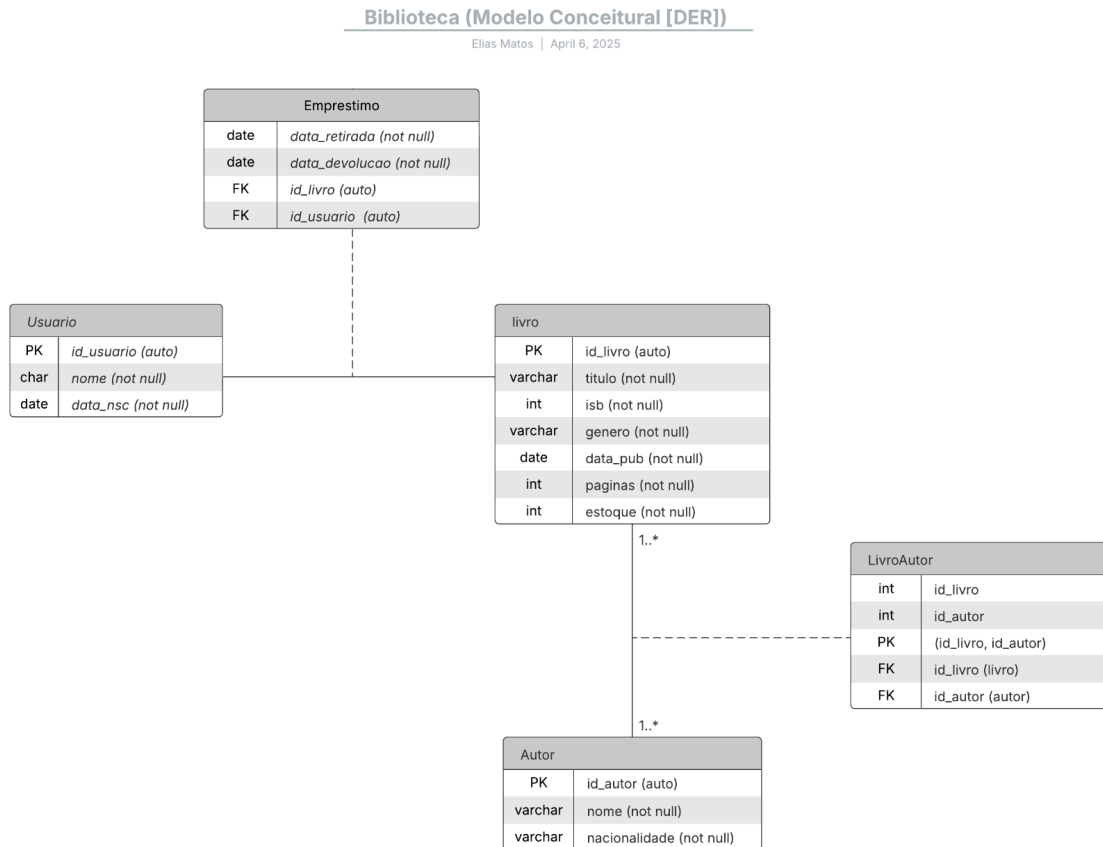
Elias Matos | April 5, 2025



### Relacionamentos:

- **Livro - Autor:** N:N → Tabela Associativa **LivroAutor**
  - *id\_livro* (FK)
  - *id\_autor* (FK)
- **Usuário - Empréstimo:** 1:N
- **Livro - Empréstimo:** 1:N

## 2. Modelo Lógico (Tabela com tipos e restrições)



```
CREATE TABLE Autor (  
    id_autor INTEGER PRIMARY KEY,  
    nome TEXT NOT NULL,  
    pais_origem TEXT NOT NULL  
);
```

```
CREATE TABLE Livro (  
    id_livro INTEGER PRIMARY KEY,  
    titulo TEXT NOT NULL,  
    isbn TEXT NOT NULL UNIQUE,  
    genero TEXT NOT NULL,  
    data_publicacao DATE NOT NULL,  
    num_paginas INTEGER NOT NULL,  
    num_exemplares INTEGER NOT NULL  
);
```

```
CREATE TABLE LivroAutor (  
    id_livro INTEGER,  
    id_autor INTEGER,  
    PRIMARY KEY (id_livro, id_autor),  
    FOREIGN KEY (id_livro) REFERENCES Livro(id_livro),  
    FOREIGN KEY (id_autor) REFERENCES Autor(id_autor)  
);
```

```

CREATE TABLE Usuario (
    id_usuario INTEGER PRIMARY KEY,
    nome TEXT NOT NULL,
    sobrenome TEXT NOT NULL,
    data_nascimento DATE NOT NULL
);

CREATE TABLE Emprestimo (
    id_emprestimo INTEGER PRIMARY KEY,
    id_usuario INTEGER NOT NULL,
    id_livro INTEGER NOT NULL,
    data_emprestimo DATE NOT NULL,
    data_devolucao DATE,
    FOREIGN KEY (id_usuario) REFERENCES Usuario(id_usuario),
    FOREIGN KEY (id_livro) REFERENCES Livro(id_livro)
);

```

### 3. — Web Scraping da página de livros

 URL: <https://pedrovnscs.github.io/livrariapython/livros.html>

Extrair da página os seguintes dados para cada livro:

- Título
- ISBN
- Gênero
- Data de publicação
- Número de páginas
- Quantidade de exemplares
- Autor(es)
- País(es) do(s) autor(es)

scraping\_livros.py

### 4. — Criação dos DataFrames:

- `df_livros`: com os dados estruturados para a tabela `Livro`
- `df_autores`: com dados únicos de autores
- `df_livro_autor`: com os relacionamentos entre livros e seus autores

transformar\_dados.py

## 5. — Criação do Banco de Dados e das Tabelas

```
CREATE TABLE IF NOT EXISTS Livro (  
    id_livro INTEGER PRIMARY KEY,  
    titulo TEXT NOT NULL,  
    isbn TEXT NOT NULL UNIQUE,  
    genero TEXT NOT NULL,  
    data_publicacao TEXT NOT NULL,  
    num_paginas INTEGER NOT NULL,  
    num_exemplares INTEGER NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS Autor (  
    id_autor INTEGER PRIMARY KEY,  
    nome TEXT NOT NULL,  
    pais_origem TEXT NOT NULL  
);  
  
CREATE TABLE IF NOT EXISTS LivroAutor (  
    id_livro INTEGER NOT NULL,  
    id_autor INTEGER NOT NULL,  
    PRIMARY KEY (id_livro, id_autor),  
    FOREIGN KEY (id_livro) REFERENCES Livro(id_livro),  
    FOREIGN KEY (id_autor) REFERENCES Autor(id_autor)  
);
```

banco.py

## 6. — Inserção dos Dados no Banco

Inserir os dados contidos nos CSVs:

- `df_livros_formatado.csv` → tabela `Livro`
- `df_autores.csv` → tabela `Autor`
- `df_livro_autor.csv` → tabela `LivroAutor`

inserir\_dados.py

## 7 — Registro de Exceções + Relatório de Erros (opcional)

- Criar uma **tabela `ErrosScraping`** no banco para registrar falhas durante o scraping (como manda o enunciado)
- Inserir nessa tabela toda exceção ocorrida ao tentar extrair algum livro
- Essa etapa pode ser **simples**: só registra a falha com descrição, livro afetado e timestamp

inserir\_dados.py