

# Sistema de Gestão de Biblioteca Universitária

## Contexto

A biblioteca da Universidade AAZZ enfrentava várias dificuldades de empréstimos de seus livros e de uma forma eficiente e acabava não gerenciando bem o seu acervo. O controle era feito a mão, onde uma única pessoa era a responsável e isso gerava:

- Perda de exemplares
- Exemplares não registrados
- Dificuldades em identificar os livros mais cobiçados
- Atrasos nas devoluções e sem cálculos automáticos para as multas
- Impossibilidade de consultar na hora se tinha o livro disponível ou não
- Estudantes cadastrados incorretamente e muitas duplicidades

## Solução Proposta

Foi desenvolvido um sistema relacional em T-SQL para automatizar e estruturar todo o fluxo da gestão da biblioteca, contemplando os livros, autores, exemplares, os membros acadêmicos e o histórico de empréstimos, com as regras de negócio já aplicadas no banco de dados garantindo integridade e consistência.

Todo o modelo foi construído usando apenas recursos nativos do SQL Server, sem dependências de ferramentas de visualização.

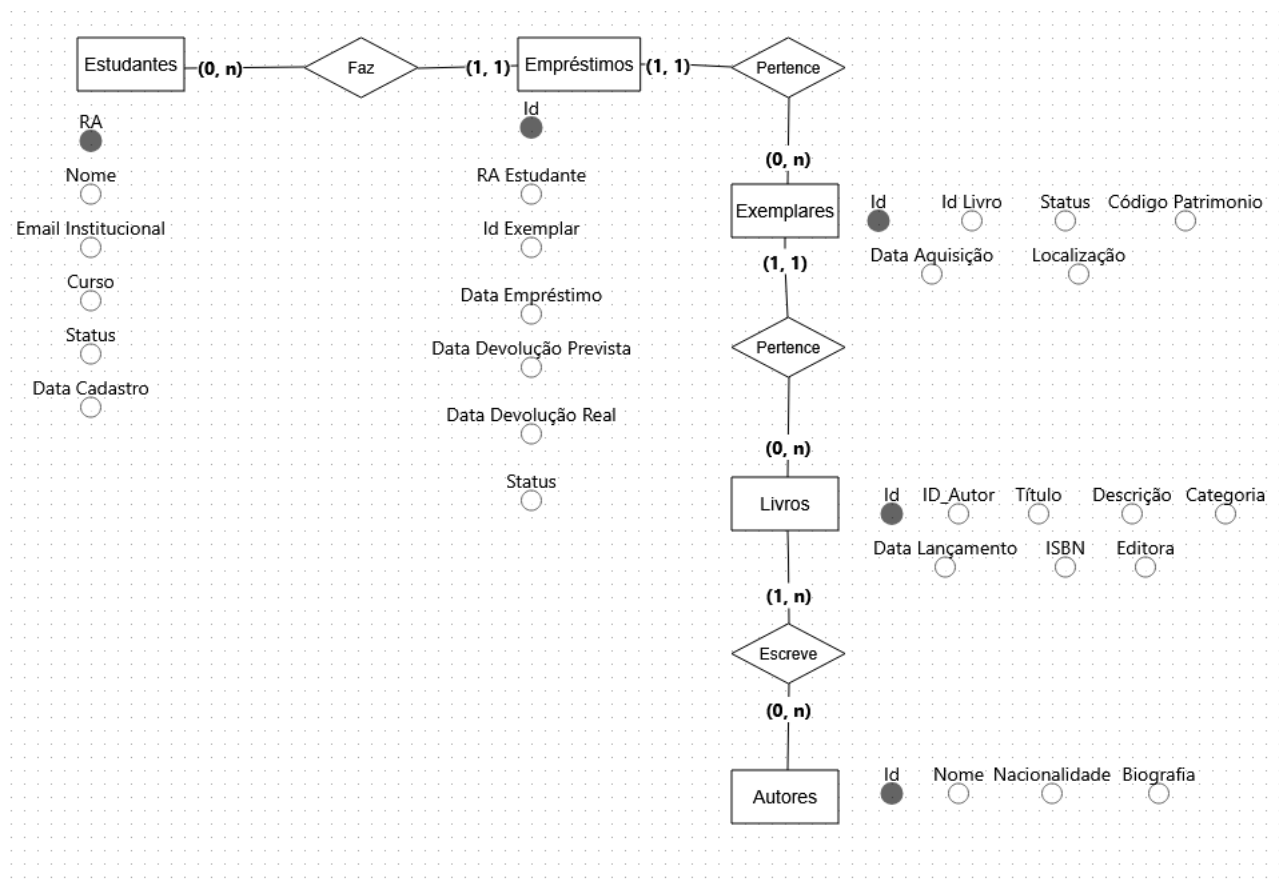
## Resultados Esperados

Com esse novo sistema, a biblioteca passa a contar com:

- Controle de status de cada empréstimo
- Controle de livros e exemplares disponíveis
- Gestão de estudantes inscritos para não se ter duplicidade
- Gestão de multas por atraso com base em regras claras
- Base sólida para futuras integrações (como um site)

Onde a coordenação terá mais controle de quem realiza os empréstimos de livros e os paga corretamente; Os estudantes terão mais controle de quando podem devolver sem pagar multas incorretas e de quando estarão disponíveis para empréstimos os livros desejados. Tudo isso com mais transparência e eficiência para toda a comunidade acadêmica.

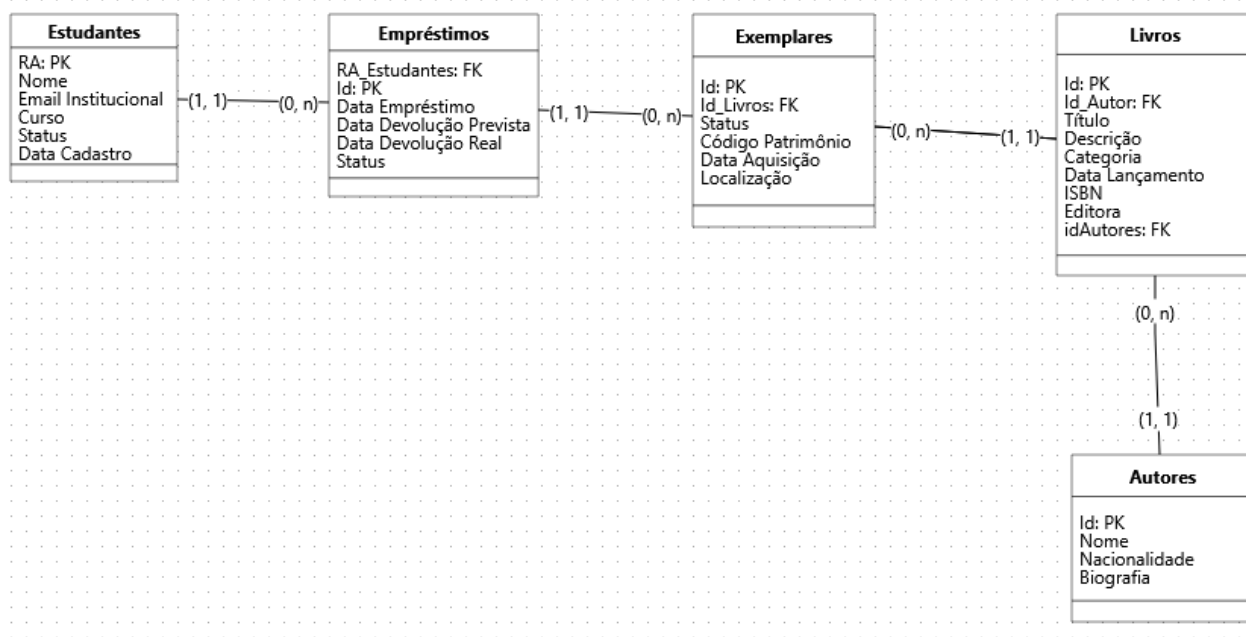
# Modelo Conceitual do Banco de Dados



## Explicação das Cardinalidades:

- Estudantes (0,n) FAZ (1,1) Empréstimos: Cada empréstimo é realizado por um único estudante, mas um estudante pode realizar vários empréstimos ao longo da vida universitária
- Exemplares (0,n) É EMPRESTADO (0,n) Empréstimo: Um exemplar pode ser emprestado várias vezes (ou nunca sem emprestado), porém cada empréstimo envolve um exemplar
- Livros (0,n) POSSUI (1,1) Exemplares: Cada exemplar pertence a um único livro, entretanto um livro pode ter nenhum ou vários exemplares na biblioteca
- Autores (0,n) ESCRIVE (1,n) Livros: Um autor pode escrever vários livros (e ter nenhum na biblioteca), mas cada livro tem que ter pelo menos um autor

# Modelo Lógico do Banco de Dados



## Tradução do modelo conceitual para modelo lógico

- Livros separados de Exemplares, pois livros podem possuir vários exemplares com uma cópia única cada um
- Tabela de empréstimos é tratada como entidade forte, pois liga estudantes e exemplares de forma controlada, onde registra datas, garante rastreabilidade completa de quem realizou um empréstimo
- Chaves estrangeiras (FK) garantem autenticidade que empréstimos sejam realizados por alunos cadastrados e por exemplares livres de uso, onde um exemplar pertença a um livro e que um livro seja vinculado a um autor existente

## Regras de Negócio Implementadas

Com as regras de negócios bem definidas, elas traduzem diretamente o que a biblioteca necessitava em estruturas de tabelas e lógica, assegurando consistência e fidelidade em todas as operações. As regras são as seguintes:

1. **Monitoramento de empréstimos em atraso:** o sistema identifica quantos estudantes estão com devoluções pendentes além do prazo.
2. **Análise de popularidade:** é possível identificar os livros mais emprestados, auxiliando na gestão do acervo e compras futuras dos livros e exemplares.
3. **Controle de disponibilidade em tempo real:** o sistema informa quantos exemplares de cada livro estão atualmente disponíveis para empréstimo.
4. **Avaliação de uso por categoria:** permite analisar a taxa de circulação de livros de tecnologia (ou outras categorias) nos últimos 6 meses.

5. **Identificação de usuários inativos:** é possível listar estudantes cadastrados que nunca realizaram um empréstimo, apoiando ações de engajamento para os estudantes.
6. **Gestão do estado físico do acervo:** o sistema rastreia exemplares em manutenção, perdidos ou indisponíveis.
7. **Métrica de comportamento dos usuários:** calcula a média de dias que os estudantes levam para devolver os livros, ajudando a ajustar políticas de prazo.
8. **Visão completa de empréstimos ativos:** lista todos os empréstimos em aberto com dados do estudante, livro, datas e status.
9. **Perfil do acervo por nacionalidade do autor:** permite quantificar e listar livros de autores brasileiros, apoiando iniciativas culturais.
10. **Análise por curso acadêmico:** identifica quais cursos têm maior engajamento com a biblioteca.
11. **Relatório abrangente de situação de empréstimos:** fornece uma visão clara do status de todos os empréstimos, orientando ações da secretaria para ter cobranças de multas corretas.

## Consultas em T-SQL

As consultas realizadas em T-SQL, fazem com que o sistema responda rapidamente às principais demandas que a biblioteca necessitava. As que mais se destacam são:

- **Situação atual de todos os empréstimos:** Nesta consulta, mostra o status de cada empréstimo e orienta qual ação a secretaria tem que tomar.

```
SELECT
    em.RaEstudante AS 'RA',
    es.Nome AS 'Estudante',
    em.DataEmprestimo AS 'Data do Empréstimo',
    em.DataDevolucaoPrevista AS 'Data Prevista de Devolução',
    em.DataDevolucaoReal AS 'Data Real de Devolução',
    CASE
        WHEN DataDevolucaoReal IS NULL THEN
            CASE -- Empréstimos abertos
                WHEN GETDATE() > DataDevolucaoPrevista THEN 'Em atraso - Procure a
Secretaria para regularizar a sua situação'
                ELSE 'Em dia (não devolvido)'
            END
        WHEN DataDevolucaoReal <= DataDevolucaoPrevista THEN
            CASE -- Empréstimos devolvidos
                WHEN DataDevolucaoReal < DataDevolucaoPrevista THEN 'Devolvido
antes do prazo'
                ELSE 'Devolvido no prazo'
            END
        WHEN DataDevolucaoReal > DataDevolucaoPrevista THEN 'Devolvido com atraso
- Procure a Secretaria para regularizar a sua situação'
        ELSE 'Situação não identificada'
    END AS 'Status do Empréstimo'
FROM Emprestimos em INNER JOIN Estudantes es ON em.RaEstudante = es.RA;
```

- **Livros mais emprestados:** Nesta consulta, identifica os títulos mais procurados pelos estudantes, apoiando decisões de aquisição de novos exemplares desses livros.

```
SELECT l.Titulo AS 'Titulo',  
       l.Categoria AS 'Categoria',  
       COUNT(*) AS 'Total de Empréstimos'  
FROM Livros l INNER JOIN Exemplares ex ON l.Id = ex.IdLivro  
INNER JOIN Emprestimos em ON ex.Id = em.IdExemplar  
GROUP BY l.Titulo, l.Categoria ORDER BY COUNT(*) DESC;
```

- **Estudantes com empréstimos em atraso:** Nesta consulta, lista e quantifica os estudantes irregulares para acompanhamento da secretaria.

```
SELECT es.Nome AS 'Nome',  
       em.RaEstudante AS 'RA',  
       em.Status AS 'Status'  
FROM Estudantes es INNER JOIN Emprestimos em ON em.RaEstudante = es.RA  
WHERE em.Status = 'Atrasado' ORDER BY es.Nome;
```

As demais consultas, estão disponíveis em meu [GitHub](#), no arquivo *Consultas.sql*.

## Próximos Passos

O sistema já cumpre o seu objetivo principal: gerenciar os empréstimos e controlar os atrasos recorrentes, tudo isso em código nativo em T-SQL. Meu pensamento como próximos passos, seria de automatizar regras de negócio contabilizando os dias de atraso e mostrando o quanto o estudante teria de pagar de multa e de aprimorar mais a validação de disponibilidade dos exemplares na hora do empréstimo.