UNIVERSIDADE DE BRASÍLIA

CIC0097 - BANCOS DE DADOS

# Projeto Final


| *Grupo 02:* | *Matrícula:* |
|---|---|
| Eduardo Afonso | 221033920 |
| Giulia Moura | 200018795 |
| Matheus Trajano | 170152227 |
| Pedro Victor | 170113043 |
| Pedro Vinícius | 211000200 |


14 de dezembro de 2023

# Conteúdo

# 1 Introdução

## 1.1 Descrição do projeto

Sistema de informação para gerenciar livros e materiais didáticos em um laboratório, com o objetivo de auxiliar estudantes e professores. Este sistema é desenvolvido para organizar e disponibilizar recursos por meio de um sistema computacional, com ênfase na definição do banco de dados que armazena informações sobre esses materiais. São estabelecidos diferentes níveis de acesso para os usuários, incluindo administradores do sistema, membros do laboratório e estudantes em geral. Dessa forma, todos os usuários podem pesquisar os recursos, mas apenas membros do laboratório têm permissão para realizar empréstimos.

## 1.2 Repositório do projeto

O repositório do projeto pode ser encontrado em https://github.com/trajano7/projetoBD_grupo2 . Os comandos escritos em SQL para a criação e população do banco de dados encontra-se em *script_criacao.sql*

## 1.3 Vídeo de apresentação

Um vídeo de apresentação do projeto foi efetuado e pode ser encontrado acessando o seguinte link: https://youtu.be/rJdUyrOI0IE

# 2 Tecnologias

## 2.1 Sistema de Gerenciamento de Bancos de Dados

Para o Sistema de Gerenciamento de Bancos de Dados, o MySQL foi escolhido, tendo em consideração algumas de suas funcionalidades, sendo algumas delas:

- Código fonte aberto: por ser um software de código aberto, o MySQL é livre para uso, distribuição e modificação. Isso resulta em menor custo de implementação e flexibilidade para personalização.

- Desempenho: o MySQL é conhecido por seu desempenho eficiente, especialmente em ambientes de leitura intensiva. Ele oferece otimizações e recursos para garantir a rápida execução de consultas.

- Escalabilidade: o MySQL é escalável e pode lidar com grandes volumes de dados e tráfego, tornando-o adequado para uma variedade de aplicações, desde pequenos projetos até sistemas empresariais de grande porte.

- Compatibilidade: O MySQL é compatível com várias plataformas e sistemas operacionais, incluindo Windows, Linux e MacOS. Isso proporciona flexibilidade na escolha do ambiente de implementação.

- Segurança: oferece recursos avançados de segurança, como suporte para SSL, criptografia de dados e capacidade de definir privilégios granulares. Isso ajuda a garantir a integridade e confidencialidade dos dados armazenados.

- Compatível com várias linguagens de programação: o MySQL é compatível com várias lingaugens de programação, como PHP, Python, Java, entre outras, facilitando a integração com diferentes aplicativos e ambientes de desenvolvimento.

- Comunidade: O MySQL possui uma comunidade de usuários ativos e uma vasta quantidade de recursos online, incluindo fóruns, tutoriais e documentação. Isso facilita a resolução de problemas e induz o aprendizado contínuo.

## 2.2 Front-end

Foi usado *HTML* e *CSS*, juntamente ao React, uma biblioteca do do *JavaScript* para a construção de interfaces de usuário (UI) interativas e dinâmicas.

## 2.3 Back-end

A linguagem de programação *Python* foi usada para o desenvolvimento dessa fase do projeto, mais especificamente, o módulo Flask, uma framework leve para desenvolvimento web do Python. Ele é projetado para ser simples, fácil de usar e flexível, permitindo que os desenvolvedores construam aplicações web de forma rápida e eficiente.
O driver oficial da Oracle, mysql-connector-python, também foi utilizado. Ele serve para conectar aplicativos Python a servidores MySQL, fornecendo uma interface Python para

interagir com um banco de dados MySQL usando a API de banco de dados padrão do Python, conhecida como DB-API.

# 3 Modelagem do banco de dados

A modelagem de banco de dados é uma prática fundamental no desenvolvimento de sistemas de informação e envolve a criação de representações estruturadas dos dados e das relações entre eles. Existem alguns princípios fundamentais na modelagem de dados, e eles ajudam a garantir a eficiência, a integridade e a flexibilidade dos sistema.

## 3.1 Modelo Entidade-Relacionamento

O modelo entidade-relacionamento (MER) é uma abordagem gráfica para modelar dados, sendo especialmente usada na fase de design de um banco de dados. Ele representa as entidades (objetos do mundo real) e os relacionamentos entre elas. As entidades são representadas por retângulos, e os relacionamentos são representadas por linhas conectando as entidades. Algumas das vantagens de se utilizar o MER são:

- Design independente do SGBD: O MER permite que os designers se concentrem na estrutura lógica dos dados sem se preocupar com os detalhes de implementação.
- Reusabilidade: Pode ser adaptado para atender a diferentes contextos, promovendo uma abordagem modular.
- Detecção de erros: Facilita a revisão e validação do modelo, minimizando erros no banco de dados final.
- Simplicidade e clareza: Evita detalhes técnicos desnecessários, concentrando-se nos aspectos essenciais do modelo.
- Abstração e visualização: Facilita a compreensão dos requisitos do sistema e das relações entre os dados.

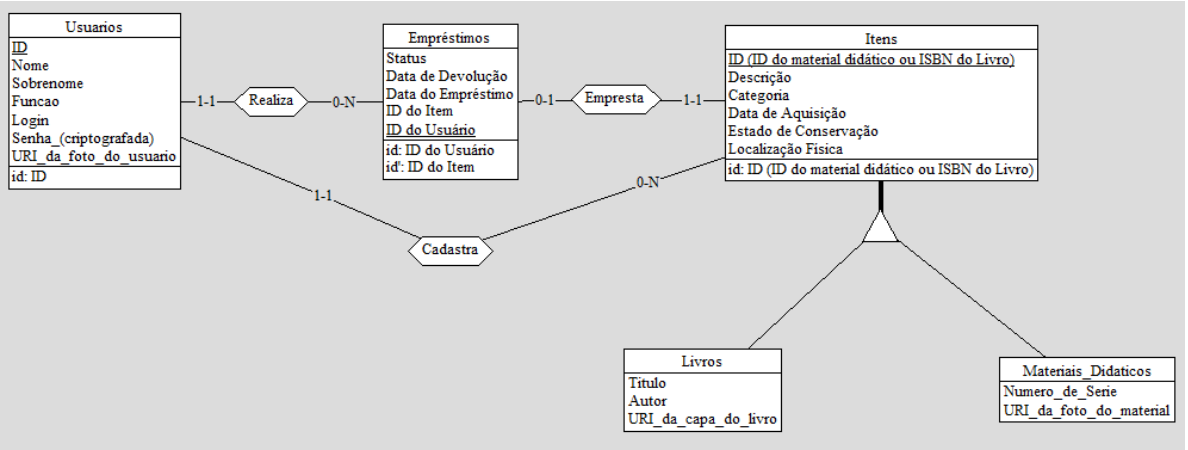


Figura 1: Modelo Entidade-Relacionamento.

Para melhor visualização da imagem, a figura 1 pode ser encontrada em */MER.png*

A descrição entidade-atributo para cada entidade é:

- Usuarios: ID, nome, sobrenome, funcao, login, senha, foto do usuário;
- Empréstimos: status, data de devolução, data do empréstimo, ID do item, ID do usuário;
- Itens: ID, descrição, categoria, data de aquisição, estado de conservação, localização física;
- Livros: Títulos, autor, foto da capa do livro;
- Materiais didáticos: Número de série, foto do material;

## 3.2 Modelo Relacional

A O Modelo Relacional (MR) é uma abordagem de modelagem de dados que utiliza tabelas para representar entidades e seus atributos. As tabelas são organizadas de forma relacional, onde as linhas representam registros individuais e as colunas representam atributos desses registros. Cada tabela possui uma chave primária única que identifica de maneira exclusiva cada registro. Relacionamentos entre tabelas são estabelecidos por meio de chaves estrangeiras. Algumas das vantagens de se usar o MR são:

- Estrutura simples e intuitiva: A estrutura baseada em tabelas é fácil de entender e representa um formato intuitivo para muitos tipos de dados.
- Integridade referencial: A integridade referencial é mantida através do uso de chaves primárias e estrangeiras, garantindo a consistência dos dados entre tabelas relacionadas.
- Flexibilidade e escalabilidade: Facilidade de expansão do modelo para incluir novas tabelas e relacionamentos, proporcionando flexibilidade para acomodar requisitos futuros.
- Normalização: A normalização ajuda a reduzir a redundância e a inconsistência nos dados, mantendo o banco de dados eficiente e evitando anomalia.

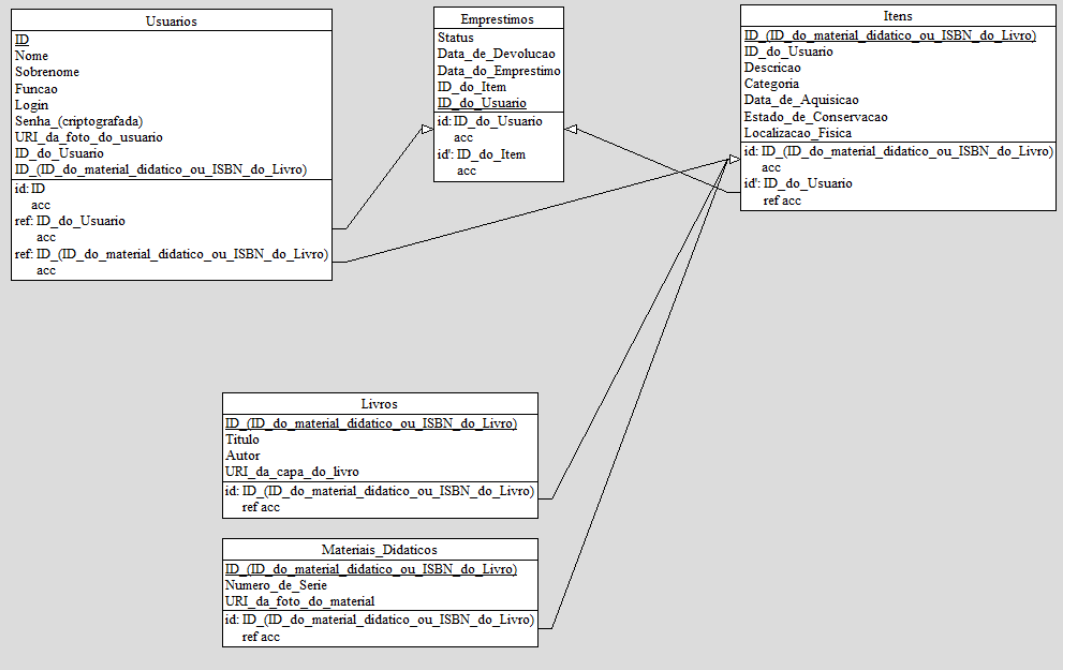


Figura 2: Modelo Relacional

Para melhor visualização da imagem, a figura 2 pode ser encontrada em */MR.png*

# 4 Manual de usuário

Para a execução do sistema, é preciso seguir alguns passo a passos. É importante lembrar que o sistema operacional utilizado foi o Linux.

## 4.1 Execução do Front-end

1. Instalar o NODE.js
    - curl -fsSL https://deb.nodesource.com/setup_lts.x | sudo -E bash -sudo apt-get install -y nodejs
2. Clonar o repositório e ir para o front-end:
    - git clone https://github.com/trajano7/projetoBD_grupo2.git cd projetoBD_grupo2/code/frontend/projeto-banco-de-dados
3. Instalar os modulos node e os pacotes no projeto
    - npm install
    - npm install router
    - npm install redux
    - npm install @reduxjs/toolkit
4. Iniciar o servidor de desenvolvimento
    - npm run dev

## 4.2 Execução do Back-end

1. Acessar e rodar um banco MySQL, seja pelo próprio MySQL ou o Workbench, phpmyadmin, etc.
2. Rodar nesse banco o script_criacao.sql encontrado no repositório
3. No arquivo app.py encontrado em /code/backend, alterar a variável db_config para as configurações de banco utilizado
4. Instalar o Python e o pip
5. Instalar as dependências do Python com um dos seguintes códigos (varia de acordo com a versão do pip da máquina utilizada):
    - pip install Flask mysql-connector-python Flask-Bcrypt flask-cors

ou

- pip3 install Flask mysql-connector-python Flask-Bcrypt flask-cors

6. Após instaladas as dependências, navegar para a pasta code/backend utilizando o terminal

7. Na pasta do backend, rodar um dos seguintes comandos (varia de acordo com a versão do python da máquina utilizada):

   - Python app.py

   ou

   - Python3 app.py

# 5 Relatório de implantação

Ao imaginar como esse projeto poderia ser implementado em uma situação da vida real, pode-se levar o seguinte cenário em consideração:

- **Laboratório de Ciências em uma Universidade**
- **Usuários:**
  - **Administradores:** São responsáveis por gerenciar os sistemas, adicionar novos livros e materiais, além de cadastrar usuários e monitorar o uso do sistema.
  - **Membros do laboratório:** Podem solicitar empréstimos, registrar a devolução e gerenciar o estado de disponibilidade dos materiais do laboratório.
  - **Estudantes em geral:** Têm acesso apenas para pesquisa de livros e materiais disponíveis, mas não podem realizar empréstimos
- **Funcionalidades:**
  - **Pesquisa de livros e materiais:** Estudantes e membros do laboratório podem procurar por livros e materiais específicos no sistema
  - **Empréstimo e devolução:** Membros do laboratório podem solicitar empréstimos, indicando a data de devolução planejada. O sistema registra empréstimos e atualiza automaticamente a disponibilidade do item.
  - **Controle de acesso:** Apenas membros do laboratório têm permissão para realizar empréstimos. Os administradores têm acesso total ao sistema.
- **Benefícios:**
  - **Organização:** O sistema ajuda a manter o inventário do laboratório organizado, facilitando a localização de recursos.
  - **Eficiência:** Processos manuais de controle de empréstimos são substituídos por um sistema automatizado, economizando tempo e minimizando erros humanos.
  - **Controle de acesso:** A implementação de diferentes níveis de acesso garante que apenas usuários autorizados possam realizar ações específicas, mantendo a segurança das informações.

Assim, o projeto se torna uma ferramenta valiosa para otimizar a gestão de recursos em

um laboratório acadêmico, proporcionando uma experiência eficiente e organizada para

estudantes e membros do laboratório.

# 6 Relatório de correções

Ao longo do desenvolvimento do Projeto, notou-se a necessidade de alteração do Framework Django, devido a algumas dificuldades enfrentadas de comunicação entre Front-end e Back-end. Dessa forma, foi escolhido um novo Framework, o Flask. A escolha do Flask para a seção de back-end em nossa aplicação foi motivada pela sua abordagem leve e modular, proporcionando flexibilidade e controle personalizado sobre os componentes utilizados. O Flask é reconhecido por sua simplicidade e facilidade de aprendizado, o que acelera o desenvolvimento e facilita a manutenção do código. Sua estrutura minimalista permite adaptar a aplicação de acordo com as necessidades específicas do projeto, resultando em um sistema eficiente e otimizado. Além disso, o ecossistema extenso de extensões do Flask oferece funcionalidades adicionais prontas para uso, contribuindo para uma implementação rápida e eficaz de recursos essenciais. Essa escolha estratégica visa otimizar o processo de desenvolvimento, garantindo um equilíbrio entre simplicidade, desempenho e flexibilidade para atender às demandas específicas do nosso projeto.