



23/11/2025

Documentação Sistema de Gestão de Biblioteca Universitária

Ambiente de dados

Sistema de Gestão de Biblioteca Universitária

Introdução



O Sistema de Gestão de Biblioteca Universitária é uma aplicação web (implementada com JavaScript/HTML/CSS para o Frontend) focada na digitalização do fluxo de trabalho de uma biblioteca. O sistema é organizado em módulos de Usuários, Livros e Empréstimos, projetado para oferecer uma interface de usuário simples e organizada.

Módulo	Objetivo	Entidades CRUD
Usuários	Gerenciar o cadastro de pessoas autorizadas a realizar empréstimos (Ex: alunos com matrícula).	Usuário (id_usuario, nome, matrícula, email).

Sistema de Gestão de Biblioteca Universitária

Módulo	Objetivo	Entidades CRUD
Livros	Manter e catalogar o acervo de títulos da biblioteca.	Livro (id_livro, título, autor, categoria, ano_publicação).
Empréstimos	Controlar a transação de retirada, a data de devolução prevista, e o registro da devolução, incluindo o cálculo de multas por atraso.	Empréstimo (id_emprestimo, data_retirada, data_devolucao, multa).

Funcionamento do Banco de Dados

Para visualizar as informações cadastradas no sistema, é necessário acessar o **MySQL** e utilizar o comando:

```
SELECT * FROM nome_da_tabela;
```

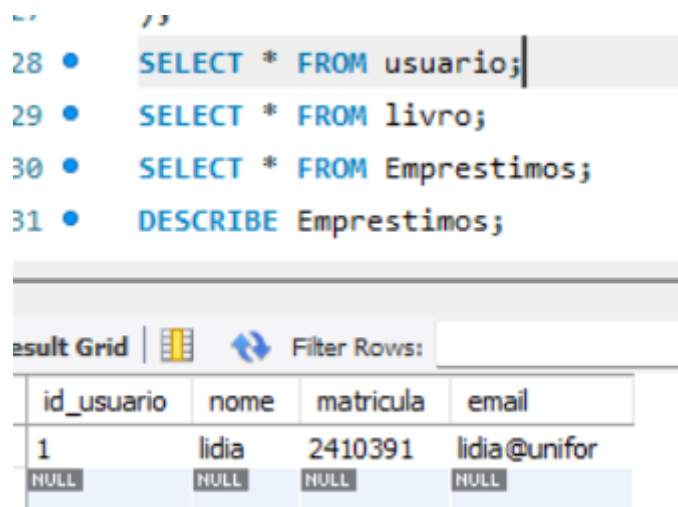
Por exemplo:

```
SELECT * FROM aluno;
```

Ao executar essa linha, o MySQL exibirá todos os registros salvos na tabela **aluno** — como o nome, CPF e data de ingresso de cada aluno.

O mesmo pode ser feito para as outras tabelas:

Sistema de Gestão de Biblioteca Universitária



Esses comandos permitem verificar se os dados cadastrados pelo sistema estão sendo salvos corretamente no banco de dados.

Dessa forma, cada ação feita nas telas (como cadastrar, atualizar, excluir) é refletida diretamente no banco de dados, garantindo a integridade e consistência das informações.

Guia Passo a Passo para um Desenvolvedor Iniciar do Zero:

Abaixo descrevo a sequência que segui no desenvolvimento, explicando as classes-chave criadas em cada etapa e como elas se relacionam.

Passo a Passo: Instalação do Visual Studio Code

1. Baixar o Instalador

1. **Acesse o Site Oficial:** Abra seu navegador e vá para o site oficial do VS Code: <https://code.visualstudio.com/>.
2. **Identifique a Versão:** O site geralmente detecta automaticamente seu sistema operacional (Windows,

Sistema de Gestão de Biblioteca Universitária

macOS ou Linux) e exibe um grande botão de **Download** para a versão correta.

3. **Inicie o Download:** Clique no botão de **Download** para salvar o instalador no seu computador.
-

2. Instalação no Windows

1. **Execute o Arquivo:** Localize o arquivo que você baixou (geralmente chamado de VSCodeSetup-x.x.x.exe) e execute-o com um duplo clique.
2. **Contrato de Licença:** Leia e aceite o contrato de licença (Licença MIT).
3. **Local de Instalação:** Mantenha o local de instalação padrão (recomendado).
4. **Opções Adicionais (Cruciais):** Esta etapa é importante. Certifique-se de marcar as seguintes caixas:
 - **Adicionar 'Abrir com Code' no menu de contexto de arquivos.**
 - **Adicionar 'Abrir com Code' no menu de contexto de diretórios.**
 - **Adicionar ao PATH** (isso permite que você inicie o VS Code de qualquer terminal, digitando code .).
5. **Concluir:** Clique em **Instalar** e, após a conclusão, clique em **Concluir** (e opcionalmente, inicie o VS Code).

Baixando MySQL Workbench

Sistema de Gestão de Biblioteca Universitária

1. Para baixar o MySQL, você precisa acessar o site oficial do MySQL. Lá, selecione a versão do MySQL que deseja instalar e baixe-a. É recomendável baixar a versão mais recente disponível.
Site oficial: <https://dev.mysql.com/downloads/workbench/>
2. Após o download, você deve iniciar o processo de instalação do MySQL. Siga as instruções na tela e aceite os termos e condições da licença. É importante selecionar as opções de configuração apropriadas durante a instalação.

Configurando o Projeto

Após concluir a instalação de todas as dependências e configurar o banco de dados, podemos iniciar o projeto para executar o sistema localmente.

Esta seção orienta o desenvolvedor sobre como configurar o ambiente de dados e inicializar a aplicação no ambiente local, utilizando comandos do terminal.

1. Pré-requisitos e Verificação de Ambiente

Antes de iniciar, o desenvolvedor deve garantir que os seguintes softwares estão instalados e acessíveis via terminal:

Software	Comando de Verificação
Node.js	node -v
NPM (Gerenciador de Pacotes)	npm -v
MySQL Server	mysql --version

2. Configuração do Banco de Dados (MySQL)

O banco de dados deve ser criado antes de a API ser executada.

1. Acessar o Terminal do MySQL:

Bash

Conecte-se ao seu servidor MySQL (usuário root é um exemplo)

mysql -u root -p

O terminal solicitará a senha

Sistema de Gestão de Biblioteca Universitária

2. Criar o Schema e as Tabelas:

SQL

-- Criar o banco de dados conforme a documentação

```
CREATE DATABASE biblioteca_universitaria;
```

```
USE biblioteca_universitaria;
```

-- Se você tiver um arquivo SQL com as definições de tabela:

```
SOURCE caminho/para/script_tabelas.sql;
```

-- Sair do terminal MySQL

```
EXIT;
```

3. **Ajustar Credenciais:** Confirme se o arquivo de conexão da API (Ex: `db_config.js` ou similar) está configurado com o usuário e senha corretos do seu MySQL local.

3. Instalação das Dependências (API Backend)

Todas as bibliotecas e *drivers* (como o Express e o driver MySQL) são instalados via npm.

1. **Navegar até o Diretório do Projeto:** O terminal deve estar na pasta que contém o arquivo `package.json` (geralmente a pasta raiz do projeto ou a pasta `/api/`).

Bash

Exemplo de navegação:

```
cd caminho/para/seu/projeto/biblioteca-universitaria
```

2. **Instalar os Pacotes NPM:** Execute o comando para baixar e instalar todas as dependências do projeto.

Bash

```
npm install
```

Aguarde a conclusão. A pasta `'node_modules'` será criada.

Sistema de Gestão de Biblioteca Universitária

4. Inicialização do Servidor de Aplicação

Após a instalação das dependências, o servidor Backend pode ser iniciado.

1. **Iniciar o Servidor:** Use o comando configurado no package.json ou chame o arquivo principal.

Bash

- OPÇÃO 1: Usando o script de inicialização definido no package.json
npm start

- OPÇÃO 2: Chamando o arquivo principal do servidor diretamente (Exemplo)

node server.js

- OU

node app.js

2. **Verificar o Status:** Após a execução, o terminal deverá exibir uma mensagem de confirmação indicando a porta em que a API está rodando (Ex: Servidor Node.js rodando na porta 3000).

5. Acesso à Interface (Frontend)

Com a API rodando, acesse a aplicação pelo navegador.

1. **Acessar a URL:** Abra qualquer navegador web e digite o endereço local. Use a porta que foi exibida no terminal.
2. `http://localhost:3000`

O sistema deve carregar a **Tela Inicial**, e você poderá testar as funcionalidades de CRUD nos módulos Usuários, Livros e Empréstimos

Manuseando

Navegação Inicial (Tela Principal)

Sistema de Gestão de Biblioteca Universitária

Ao iniciar a aplicação, a tela principal oferece acesso direto aos três módulos principais:

- **Usuários:** Para gerenciamento do cadastro de alunos e funcionários.
- **Livros:** Para gerenciamento do acervo.
- **Empréstimos:** Para registrar transações de retirada e devolução.



Usuários



Sistema de Gestão de Biblioteca Universitária

 **Novo Usuário**

Nome Completo

Preencha este campo.

Matrícula

Email Institucional

Salvar Cadastro

Cancelar

Módulo Usuários é responsável por todas as operações de **Cadastro, Consulta, Edição e Exclusão** dos alunos e funcionários que farão empréstimos na biblioteca.

1. Funcionalidade de Criação (CREATE)

Esta operação é usada para incluir novos usuários na base de dados, garantindo que apenas pessoas com matrícula válida possam acessar os serviços.

Detalhamento da Funcionalidade: Criar Usuário (CREATE)

Sistema de Gestão de Biblioteca Universitária

A operação de criação de usuário é a primeira etapa para registrar novos alunos e funcionários no sistema da biblioteca, concedendo-lhes permissão para realizar empréstimos.

Fluxo de Manuseio

1. **Acesso à Funcionalidade:** O operador do sistema deve navegar até o módulo **Usuários** e clicar no botão "**Novo Usuário**" ou "**Cadastrar**" na interface.
2. **Exibição do Formulário:** O sistema exibe a tela de cadastro, solicitando os dados obrigatórios para a criação do registro.
3. **Preenchimento dos Campos:**
 - **Nome Completo:** O nome completo do aluno ou funcionário.
 - **Matrícula:** O número único de identificação institucional.
 - **Email:** O endereço de email institucional.
4. **Ação de Confirmação:** Após preencher todos os campos, o operador clica no botão "**Salvar Cadastro**".

Lógica e Validação (Backend)

No momento da submissão, a API Backend é responsável por garantir a integridade dos dados:

- **Validação de Unicidade:** É obrigatório que a API verifique se a **Matrícula** e o **Email** informados **não estão duplicados** na tabela Usuario. Isso é fundamental, pois esses campos estão definidos como **UNIQUE** no banco de dados. Caso a matrícula ou o email já exista, o sistema deve retornar uma mensagem de erro ao usuário.

Sistema de Gestão de Biblioteca Universitária

- **Backend (SQL):** Se a validação for bem-sucedida, a seguinte *query* SQL é executada para inserir os dados na base:

SQL

```
INSERT INTO Usuario (nome, matricula, email)
```

```
VALUES ('Lídia', '2410391', 'lídia@universidade.edu.br');
```

Manuseio Detalhado: Módulo Livros



O Módulo Livros é a área do sistema dedicada à gestão do acervo físico e digital da biblioteca. Ele permite que o operador realize todas as operações necessárias para manter o catálogo de títulos atualizado.

1. Funcionalidade de Criação

2. A criação é o processo de adicionar um novo título ao acervo.

Para iniciar, o operador deve acessar o Módulo Livros e clicar no botão "**Novo Livro**" ou "**Cadastrar Livro**". O sistema irá exibir um formulário onde serão solicitados os dados catalográficos da obra.

Sistema de Gestão de Biblioteca Universitária

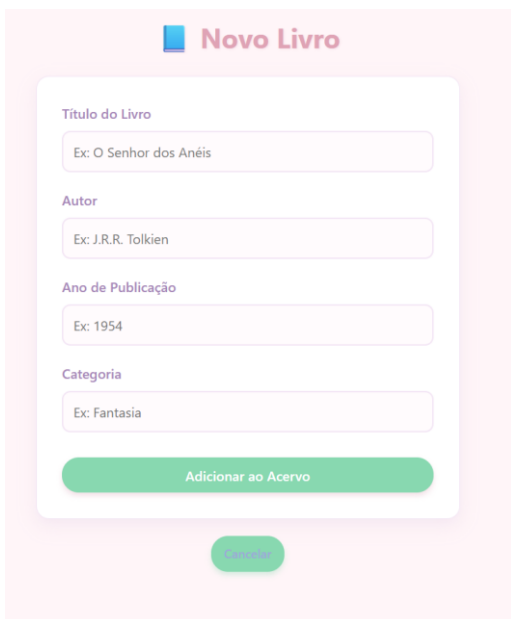
O preenchimento exige os seguintes campos: o **Título** da obra, o **Autor** principal, o **Ano de Publicação** e a **Categoria** do livro (como "Direito", "Tecnologia" ou "Literatura").

Após o preenchimento, o operador clica em "**Salvar Livro**" ou "**Adicionar ao Acervo**".

No Backend, a API valida que o Título e o Autor foram fornecidos. Se a validação for bem-sucedida, a *query* SQL é executada:

SQL

```
INSERT INTO Livro (titulo, autor, ano_publicacao, categoria)
VALUES ('Título do Livro', 'Nome do Autor', 2023, 'Categoria');
```



O formulário, intitulado "Novo Livro", está contido em uma caixa branca com sombra sobre um fundo rosa claro. Ele possui quatro campos de entrada de texto, cada um com um rótulo em roxo e um exemplo de valor em cinza: "Título do Livro" (Ex: O Senhor dos Anéis), "Autor" (Ex: J.R.R. Tolkien), "Ano de Publicação" (Ex: 1954) e "Categoria" (Ex: Fantasia). Abaixo dos campos, há um botão verde arredondado com o texto "Adicionar ao Acervo". Na base do formulário, há um botão verde menor com o texto "Cancelar".

Sistema de Gestão de Biblioteca Universitária

Manuseio Empréstimos



O Módulo Empréstimos é o centro operacional do Sistema de Biblioteca, controlando a movimentação dos livros do acervo. Este módulo utiliza a integração entre as tabelas Usuario e Livro para registrar cada transação.

1. Funcionalidade de Criação (CREATE): Novo Empréstimo

Esta operação registra a retirada de um livro por um usuário.

Acesso e Formulário: O operador acessa a tela de Controle de Empréstimos e clica no botão "+ Novo Empréstimo". Será exibido um formulário de confirmação.

Sistema de Gestão de Biblioteca Universitária

 **Registrar Empréstimo**

Quem vai retirar?

Selecione o Usuário... ▾

Qual livro?

Selecione o Livro... ▾

Data de Retirada

dd/mm/aaaa 

Data de Devolução Prevista

dd/mm/aaaa 

Confirmar Empréstimo

Cancelar

Preenchimento:

- **Usuário:** O operador seleciona o usuário (aluno ou funcionário) que está retirando o livro a partir de um *dropdown* (Ex: "Ídia Mat: 2410391").
- **Livro:** O livro a ser emprestado também é selecionado por *dropdown* (Ex: "branca de neve").
- **Data de Retirada:** A data atual da transação é registrada.
- **Data de Devolução Prevista:** Uma data futura (baseada na política da biblioteca, ex: 7 ou 15 dias) é definida e registrada.

Sistema de Gestão de Biblioteca Universitária

Lógica de Negócios (Validação Crucial): Antes de salvar, a API Backend deve aplicar três validações:

1. **Usuário e Livro Existentes:** Confirma se o `id_usuario` e o `id_livro` selecionados são válidos.
2. **Disponibilidade do Livro:** Verifica se o livro não está atualmente emprestado (ou se há cópias disponíveis, se o sistema gerenciar múltiplas cópias).
3. **Pendências do Usuário:** Impede o empréstimo se o usuário tiver multas não pagas ou um número excessivo de livros emprestados.

Ação: Ao clicar em "**Confirmar Empréstimo**", a API executa a inserção:

SQL

```
INSERT INTO Empréstimo (fk_id_usuario, fk_id_livro,
data_retirada, data_devolucao_prevista, multa)
VALUES ([ID_USUARIO], [ID_LIVRO], CURDATE(),
DATE_ADD(CURDATE(), INTERVAL 7 DAY), 0.00);
```

O campo `data_devolucao_real` é deixado como NULL.

2. Funcionalidade de Leitura : Controle de Transações

A leitura é a visualização em tempo real de todas as transações, ativas e finalizadas.

Acesso: O operador acessa o Módulo Empréstimos, onde uma **Tabela de Controle** é exibida.

Visualização: A tabela exibe informações detalhadas de cada empréstimo: **ID do Empréstimo**, **Usuário**, **Livro**, **Data de Retirada**, **Data de Devolução** (real) e o valor da **Multa** aplicada. Note que o Nome do Usuário e o Título do Livro são

Sistema de Gestão de Biblioteca Universitária

exibidos, exigindo que a API utilize comandos JOIN entre as tabelas.

Busca e Filtros: Filtros são essenciais para localizar rapidamente empréstimos por usuário, por livro ou para listar apenas **Empréstimos Ativos** (onde data_devolucao é NULL).

Backend (SQL):

SQL

SELECT

E.id_emprestimo,
U.nome AS nome_usuario,
L.titulo AS titulo_livro,
E.data_retirada,
E.data_devolucao,
E.multa

FROM Emprestimo E

JOIN Usuario U ON E.fk_id_usuario = U.id_usuario

JOIN Livro L ON E.fk_id_livro = L.id_livro;

3. Funcionalidade de Atualização (UPDATE): Processamento de Devolução

O processamento de devolução é o momento em que a transação é encerrada e a multa, se houver, é calculada e registrada.

Acesso: Na tabela de listagem (READ), o operador clica no botão "**Ação**" ou "**Devolver**" referente ao empréstimo que está sendo devolvido.

Sistema de Gestão de Biblioteca Universitária

Registro da Devolução: A API automaticamente registra a data do dia como a data_devolucao real para o registro.

Lógica de Negócios (Cálculo de Multa):

- O sistema compara a data_devolucao real com a data_devolucao_prevista.
- Se a data real for posterior à prevista, a API calcula a multa com base no número de dias de atraso e na tarifa diária estabelecida pela biblioteca.
- O campo multa é atualizado com o valor calculado.

Ação: A API executa a atualização no banco de dados, finalizando o registro de empréstimo:

SQL

UPDATE Emprestimo

SET data_devolucao = CURDATE(), multa =
[VALOR_CALCULADO_DA_MULTA]

WHERE id_emprestimo = [ID do Empréstimo];

4. Funcionalidade de Exclusão (DELETE)

A exclusão direta de um registro de empréstimo não é comum em sistemas de biblioteca, pois o registro histórico é importante. A função de "excluir" seria, na prática, um **cancelamento** antes da devolução.

Restrição: Esta operação deve ser usada apenas em casos excepcionais (ex: erro de registro imediato). O sistema deve **impedir a exclusão** de empréstimos já finalizados, ou seja, onde a data_devolucao não é NULL.

Backend (SQL):

Sistema de Gestão de Biblioteca Universitária

SQL

DELETE FROM Emprestimo

WHERE id_emprestimo = [ID do Empréstimo];

Links github:

<https://github.com/anakarolinasf/SistemaDeGestaoDeBibliotecaUniversitaria.git>

<https://github.com/anakarolinasf/SistemaDeGestaoDeAcademia.git>