

Universidade de Brasília - UnB

Departamento de Ciência da Computação



Relatório Final do Projeto
Sistema de Gerenciamento de Materiais para um Laboratório
Didático

CIC0097 - BANCOS DE DADOS - T01 - Grupo 5

Autoria:

GABRIEL SANTOS PEREIRA 140140671

LUANA PINHO TORRES 222011623

LUIS FELIPE HILARIO CARMO 212007153

RYAN REIS FONTENELE 211036132

Brasília
14 de dezembro de 2023

Conteúdo

1	Introdução	2
1.1	O que é um Banco de dados?	2
1.2	Repositório do Projeto	2
1.3	Vídeo de Apresentação	2
2	Tecnologias Utilizadas	3
3	Modelagem do Banco de Dados	5
4	Como instalar as ferramentas	7
4.1	Node	7
4.2	Express	8
4.3	MySQL	9
4.4	HTML	11
4.5	CSS	12
5	Referências	13

1 Introdução

1.1 O que é um Banco de dados?

Um banco de dados é um sistema organizado e estruturado que armazena, gerencia e fornece acesso a informações e dados. Ele é projetado para armazenar dados de forma eficiente, permitindo a recuperação, atualização e análise desses dados de maneira rápida e confiável. Os bancos de dados são amplamente utilizados em diversas aplicações e setores, incluindo negócios, ciência, governo e muitos outros, para armazenar informações estruturadas, como registros de clientes, transações financeiras, informações de produtos e muito mais. Eles desempenham um papel fundamental na organização, recuperação e análise de informações para suportar operações diárias e tomada de decisões. Existem vários tipos de sistemas de gerenciamento de banco de dados (SGBDs), como MySQL, Oracle, SQL Server e MongoDB, que oferecem funcionalidades diferentes e são escolhidos com base nas necessidades específicas de cada aplicação.

1.2 Repositório do Projeto

O repositório para o projeto de Sistema de Gerenciamento de Materiais para um Laboratório Didático se encontra na plataforma Github, que pode ser acessado por esse link: <https://github.com/LTCrash/ProjetoBancoDeDados>.

1.3 Vídeo de Apresentação

O vídeo de apresentação de uso do projeto de Sistema de Gerenciamento de Materiais para um Laboratório Didático se encontra na plataforma Youtube, que pode ser acessado por esse link: https://www.youtube.com/watch?v=chCK0CD_c3s.

2 Tecnologias Utilizadas

Neste projeto, desenvolvemos um "Sistema de Gerenciamento de Materiais para um Laboratório Didático" que consiste em uma aplicação web com duas partes principais: o "front-end" e o "back-end", juntamente com um banco de dados.

- **Front-End:** O "front-end" é a parte da aplicação com a qual os usuários interagem diretamente. Ele é responsável pela interface gráfica, layout e experiência do usuário. Inicialmente, optamos por utilizar o framework "React", uma biblioteca de JavaScript de código aberto amplamente usada para construir interfaces de usuário dinâmicas e responsivas. No entanto, após uma reavaliação, decidimos utilizar "HTML" em conjunto com "CSS". Essa decisão foi baseada em vários fatores. Primeiramente, temos uma maior familiaridade com as linguagens e estruturas do HTML e CSS. Além disso, poderíamos implementar a interface com maior eficiência, pois não precisaríamos utilizar bibliotecas adicionais para escrever o HTML, como seria necessário na implementação do front-end com "React". Finalmente, optamos pelo HTML e CSS devido à quantidade de materiais de apoio disponíveis. Como são linguagens bem consolidadas, elas apresentam suporte universal e não necessitam de dependências extras. Isso nos permite focar na criação de uma interface de usuário eficaz e agradável, sem a necessidade de gerenciar complexidades adicionais;
- **Back-End:** O "back-end" é a parte invisível da aplicação que lida com a lógica de negócios, processamento de dados e interações com o banco de dados. Para a implementação do "back-end" escolhemos o framework "Node.js", que é um ambiente de tempo de execução de JavaScript no servidor. Ele permite a criação de aplicativos web do lado do servidor usando JavaScript. Usamos nossa própria API para fornecer endpoints para o "front-end" se comunicar com o "back-end". A API define como os dados são solicitados e como as respostas são formatadas;
- **Banco de Dados (SGBD):** O Sistema de Gerenciamento de Banco de Dados (SGBD) escolhido foi o "SQLite", que é um sistema de gerenciamento de banco de dados relacional de código aberto. Ele é uma opção leve e amplamente utilizada para armazenar dados. O banco de dados é usado para armazenar e recuperar informações relacionadas aos materiais do laboratório, como inventário, registros de uso e outras informações importantes. No entanto, apesar das vantagens do SQLite, optamos pelo MySQL em nossos projetos. O MySQL é um sistema de gerenciamento de banco de dados relacional robusto e altamente escalável. Ele oferece uma série de recursos

avançados, como suporte a transações, alta disponibilidade, segurança aprimorada e uma ampla gama de tipos de dados. Além disso, o MySQL tem uma grande comunidade de usuários e desenvolvedores, o que significa que há uma abundância de documentação, tutoriais e soluções para problemas comuns disponíveis. Em resumo, embora o SQLite seja uma excelente opção para aplicações mais leves e menos complexas, o MySQL é preferido para aplicações que exigem mais robustez, escalabilidade e recursos avançados;

- No contexto de uma aplicação web, a **comunicação entre o "front-end" e o "back-end"** ocorre por meio de solicitações HTTP. O "front-end" faz solicitações para a API do "back-end" para buscar ou enviar dados. O "back-end" processa essas solicitações, realiza operações no banco de dados conforme necessário e envia respostas de volta ao "front-end". Essa conexão entre "front-end" e "back-end" permite que os usuários interajam com o sistema e acessem informações de maneira segura e eficiente. O uso de tecnologias como "React" e "Node.js" facilita o desenvolvimento ágil e escalável desse tipo de aplicação.

3 Modelagem do Banco de Dados

A criação de uma modelagem ao banco de dados é uma prática útil para visualizar e conceituar relacionamentos e entidades, conduzindo a uma melhor compreensão das funcionalidades do sistema em relação ao banco de dados.

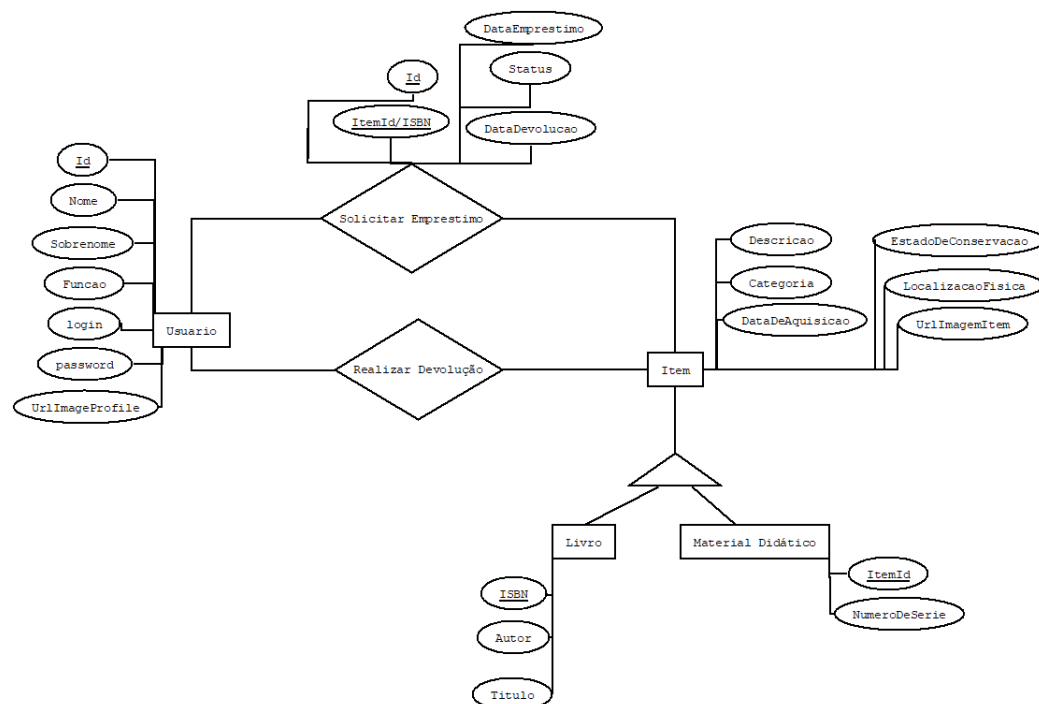


Figura 1: Modelo Entidade-Relacionamento

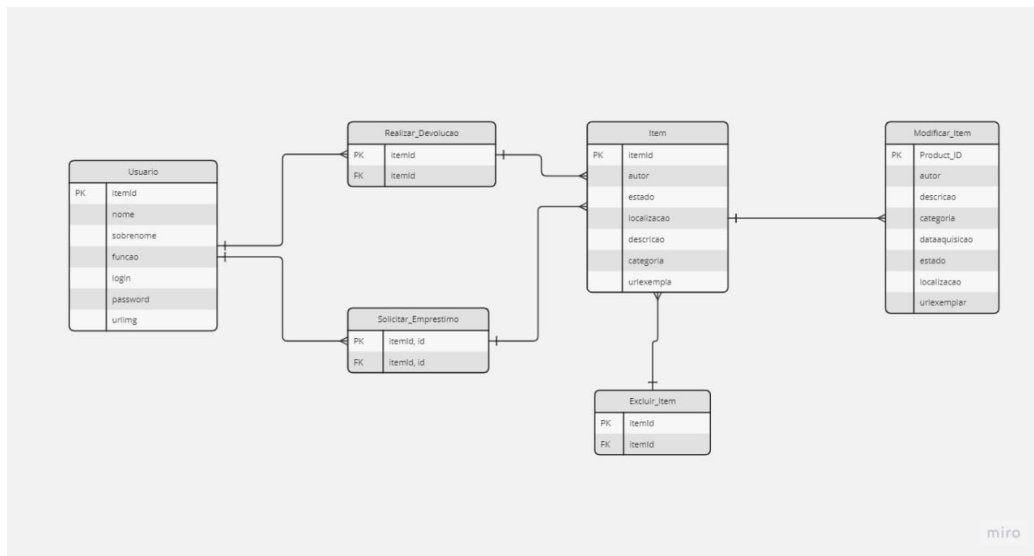


Figura 2: Modelo Relacional

1. Cadastro(PRIMARY key: itemId:int; autor: varchar, descricao: varchar, categoria: varchar, dataaquisicao: date, estado: varchar, titulo: varchar, localizacao: varchar, urlimagem: varchar) - User Autorizado
2. ModificarCadastro(PRIMARY key: itemId:int; autor: varchar, descricao: varchar, categoria: varchar, dataaquisicao: date, estado: varchar, localizacao: varchar, titulo: varchar, urlimagem: varchar, FOREIGN key: itemId) - User Autorizado
3. ExcluirCadastro(PRIMARY key: itemId:int, FOREIGN key: itemId) - User Autorizado
4. SolicitarEmprestimo(dataEmprestimo: date, dataDevolucao: date, status: varchar, PRIMARY keys: id: int, itemId: int; Foreign keys: id, itemId)
5. RealizarDevolucao(PRIMARY keys: itemId, Foreign keys: itemId)
6. Usuario(PRIMARY key id: int, nome: varchar, sobrenome: varchar, funcao: int, login:varchar, password:varchar, urlimg: varchar)

4 Como instalar as ferramentas

Uma parte muito importante do projeto são as ferramentas para executá-lo, usaremos então Node.js, MySQL, HTML, CSS e express.js como algumas ferramentas para o desenvolvimento.

4.1 Node

Aqui estão os passos resumidos para instalar o Node.js no Windows:

1. Baixe o instalador do Node.js do site oficial: <https://nodejs.org/en/>.
2. Execute o instalador clicando duas vezes no arquivo baixado.
3. Clique em "Next" na janela do instalador.
4. Leia e aceite os termos de uso marcando a caixa de seleção e clique em "Next".
5. Escolha o local de instalação padrão e clique em "Next".
6. Selecione as funcionalidades que deseja instalar (é recomendado manter as opções padrão) e clique em "Next".
7. Escolha a opção para instalar as ferramentas para módulos nativos e o Chocolatey (recomendado) e clique em "Next".
8. Clique em "Install" para iniciar a instalação.
9. Aguarde a conclusão da instalação.
10. Verifique a instalação abrindo o Prompt de Comando e digitando 'node -version' e 'npm -version'. Certifique-se de reiniciar o computador após a instalação.

Agora você tem o Node.js e o npm instalados no seu sistema Windows e já pode criar um novo diretório para o seu projeto.

4.2 Express

Express, também conhecido como Express.js, é um framework web para Node.js que simplifica e acelera o desenvolvimento de aplicações web. Ele fornece um conjunto robusto de recursos para lidar com requisições HTTP, gerenciar rotas, renderizar vistas e facilitar a construção de APIs RESTful de forma eficiente.

Para usar o Express no projeto, primeiro, devemos efetuar a instalação para Node.js. Execute o seguinte comando no terminal: "npm install express".

Crie um arquivo app.js com o seguinte conteúdo : app.js no seu editor de código e configure o Express para criar um servidor básico. Adicione o seguinte código:

```
// app.js
const express = require('express');
const app = express();
const porta = 3000;

app.get('/', (req, res) => {
  res.send('Hello, World!');
});

app.listen(porta, () => {
  console.log(`Servidor rodando em
    http://localhost:${porta}`);
});
```

Abra o navegador e acesse <http://localhost:3000>. Você deverá ver a mensagem "Hello, World!".

4.3 MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional (RDBMS) de código aberto. Ele é amplamente utilizado para armazenar e gerenciar dados em aplicações web e outros sistemas. O MySQL utiliza a linguagem SQL (Structured Query Language) para realizar operações de consulta, inserção, atualização e exclusão de dados em bancos de dados relacionais. Desenvolvido pela Oracle Corporation, o MySQL é conhecido por sua confiabilidade, desempenho e facilidade de uso, sendo uma escolha popular para uma variedade de aplicações e projetos.

Para usar o MySQL no projeto, primeiro, devemos instalar o pacote "mysql" para Node.js. Execute o seguinte comando no terminal: "npm install mysql".

Para criar a conexão com o MySQL crie um arquivo "conexão.js" para configurar a conexão e adicione o seguinte código ao arquivo:

```
// conexao.js
const mysql = require('mysql');

const conexao = mysql.createConnection({
  host: 'localhost',
  user: 'seu-usuario',
  password: 'sua-senha',
  database: 'nome-do-banco'
});

conexao.connect((err) => {
  if (err) {
    console.error('Erro ao conectar ao MySQL:', err);
  } else {
    console.log('Conectado ao MySQL!');
  }
});

module.exports = conexao;
```

Substitua "seu-usuario", "sua-senha" e "nome-do-banco" pelos detalhes da sua configuração.

Agora, você pode usar o arquivo `conexao.js` no seu projeto para realizar operações no MySQL. Por exemplo, abra o arquivo `app.js` e adicione o seguinte conteúdo:

```
// app.js
const conexao = require('./conexao');

// Exemplo: Consultar todos os usuarios
conexao.query('SELECT * FROM usuarios', (err, resultados)
=> {
  if (err) {
    console.error('Erro ao executar a consulta:', err);
  } else {
    console.log('Resultados da consulta:', resultados);
  }

  // Fechar a conexão após as consultas
  conexao.end();
});
```

4.4 HTML

HTML (Hypertext Markup Language) é a linguagem padrão para estruturar conteúdo em páginas web.

Para usar o HTML no projeto, dentro do diretório em uso, crie um arquivo HTML chamado "index.html" e adicione a estrutura básica do HTML:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
    initial-scale=1.0">
  <title>Meu Projeto</title>
</head>
<body>

  <!-- Contedo do corpo vai aqui -->

</body>
</html>
```

Dentro do bloco «body» você pode adicionar referências ao seu script JavaScript. Para isso crie um arquivo JavaScript chamado script.js e adicione o seguinte código ao arquivo script.js:

```
// script.js
console.log("Hello, world!");
```

Para conectar o JavaScript ao HTML volte ao arquivo index.html e adicione uma referência ao seu script JavaScript no final do corpo:

```
<!-- Contedo do corpo vai aqui -->

<script src="script.js"></script>
```

Agora já temos um projeto básico que inclui HTML e JavaScript.

4.5 CSS

CSS, ou Cascading Style Sheets, é uma linguagem de estilo utilizada para definir a apresentação e a aparência visual de documentos HTML ou XML em páginas web, permitindo o controle de layout, cores, fontes e outros estilos.

Para usar o CSS no projeto, dentro do diretório em uso, crie um arquivo CSS chamado "styles.css" e adicione alguns estilos simples a estrutura básica do CSS:

```
/* styles.css */

body {
    font-family: Arial, sans-serif;
    background-color: #f4f4f4;
    margin: 0;
    padding: 0;
}

h1 {
    color: #333;
    text-align: center;
    padding: 20px;
    background-color: #66b2ff;
    margin: 0;
}
```

Observe como os estilos definidos no arquivo CSS afetam a aparência da página. No exemplo acima, o fundo do corpo da página é cinza claro, e o título é estilizado com uma cor específica, alinhamento central e um fundo azul claro.

Este é só um exemplo simples, e você pode expandir e personalizar os estilos de acordo com as necessidades do seu projeto à medida que ele cresce.

5 Referências

1. V. Oviedo. "Como eu exibo blocos de código em LaTeX". Medium. Consult. 2023-09-26. [Em linha]. Disponível: <https://encurtador.com.br/fuZ46>
2. I. Araújo e R. Silvério. "Como instalar o Node.js no Windows, Linux e macOS — Alura". Alura. Consult. 2023-09-26. [Em linha]. Disponível: <https://l1nq.com/F1v5L>
3. "Express Instalação". Express Instalação Page. Consult. 2023-10-05. [Em linha]. Disponível: <https://expressjs.com/pt-br/starter/installing.html>
4. A. Almeida. "Integrando o MySQL ao NodeJS". Medium. Consult. 2023-10-10. [Em linha]. Disponível: <https://encurtador.com.br/cjBMO>
5. Andrei L. "Como Adicionar JavaScript no HTML". Hostinger. Consult. 2023-11-13. [Em linha]. Disponível: <https://www.hostinger.com.br/tutoriais/como-adicionar-javascript-no-html>
6. "Microsoft Learn". Microsoft Documentação Page. "Edição de estilo para estruturas CSS-in-JS". React. Consult. 2023-11-15. [Em linha]. Disponível: <https://learn.microsoft.com/pt-br/microsoft-edge/devtools-guide-chromium/css/css-in-js>