

Nome: Davi da Silva Becker e Lucas Amâncio

Data: 02/07/2025

# API de Gerenciamento de Cinema – CineLux

## Objetivo

O objetivo deste trabalho foi desenvolver uma API para gerenciar as principais funcionalidades de um cinema, como o cadastro de filmes, salas, sessões, clientes e ingressos. A aplicação foi construída utilizando Express como framework de servidor e MySQL como banco de dados relacional.

## Estrutura da API

A API foi dividida em rotas específicas para cada entidade do sistema:

1. Filmes  
Permite cadastrar novos filmes e listar os filmes em cartaz, com informações como título, gênero e número de ingressos disponíveis.
2. Salas  
Gerencia as salas de exibição, contendo o nome da sala e a capacidade de público.
3. Sessões  
Relaciona os filmes às salas e horários de exibição, permitindo o controle da programação.
4. Clientes  
Armazena informações dos clientes, como nome, e-mail e telefone, permitindo registrar quem está comprando ingressos.
5. Ingressos  
Registra as vendas de ingressos, associando cliente, sessão, assento e preço.

# Tecnologias Utilizadas

- Node.js: Ambiente de execução JavaScript no servidor.
- Express.js: Framework web leve e flexível para criação da API.
- MySQL: Banco de dados relacional para armazenar as informações.
- MySQL2: Driver utilizado para conectar o Node.js ao banco de dados.
- Postman: Ferramenta de testes de API (sugestão para testar os endpoints).

## Endpoints da API

Abaixo, uma visão geral dos principais endpoints:

- GET /filmes – Lista todos os filmes.
- POST /filmes – Cadastra um novo filme.
- GET /salas – Lista todas as salas.
- POST /salas – Cadastra uma nova sala.
- GET /sessoes – Lista todas as sessões.
- POST /sessoes – Cadastra uma nova sessão.
- GET /clientes – Lista todos os clientes.
- POST /clientes – Cadastra um novo cliente.
- GET /ingressos – Lista todos os ingressos vendidos.
- POST /ingressos – Registra a compra de um ingresso.

## Banco de Dados

A estrutura do banco foi criada com tabelas relacionadas para garantir integridade referencial entre os dados, como a associação de sessões a filmes e salas, e de ingressos a sessões e clientes.

# Resultados

Ao final do desenvolvimento, obtivemos uma API funcional, capaz de:

- Gerenciar a programação de filmes e salas.
- Controlar as sessões com horários e locais definidos.
- Cadastrar clientes e registrar compras de ingressos.
- Oferecer um ponto central de integração para sistemas de frontend, mobile ou até painéis administrativos.

## Conclusão

Este projeto demonstra a aplicação prática de conceitos de API REST, banco de dados relacional, manipulação de dados com SQL e organização de código com Node.js e Express. Ele pode ser expandido para incluir autenticação de usuários, filtros de busca, dashboards administrativos e muito mais.

