

# Atividade: Criando um CRUD de Tarefas com Frontend e Backend (MySQL)

## Objetivo:

Desenvolver um sistema de lista de tarefas utilizando **frontend e backend**, com armazenamento no banco de dados **MySQL**. O projeto deve permitir adicionar, visualizar, editar e excluir tarefas.

## Requisitos:

- Utilizar **Node.js, Express e MySQL** para o backend.
  - Criar um **banco de dados MySQL** para armazenar as tarefas.
  - Desenvolver um **frontend com HTML, CSS e JavaScript** para interagir com a API.
  - Implementar um **CRUD completo** (Create, Read, Update, Delete).
- 

## Passo 1: Configuração do Banco de Dados

1. Criar um banco de dados no MySQL chamado `tarefas_db`.
2. Criar uma tabela chamada `tarefas` com os seguintes campos:
  - `id` (chave primária, autoincremento)
  - `nome` (texto, obrigatório)
3. Inserir algumas tarefas de teste manualmente para validar a estrutura.

**Dica:** Utilize o MySQL Workbench ou comandos SQL para criar o banco e a tabela.

---

## Passo 2: Desenvolvimento do Backend

1. Criar um projeto Node.js e instalar as dependências necessárias: `express`, `mysql2` e `cors`.
2. Criar um arquivo `server.js` e configurar um servidor com **Express**.

3. Implementar rotas para:

- Buscar todas as tarefas ( `GET /tasks` )
- Criar uma nova tarefa ( `POST /tasks` )
- Atualizar uma tarefa ( `PUT /tasks/:id` )
- Excluir uma tarefa ( `DELETE /tasks/:id` )

4. Configurar a conexão com o **MySQL** e garantir que as requisições sejam processadas corretamente.

**Dica:** Utilize `express.json()` para permitir requisições JSON e teste as rotas com o Postman ou cURL.

---

## Passo 3: Desenvolvimento do Frontend

1. Criar um arquivo `index.html` contendo:

- Um campo de entrada para adicionar tarefas.
- Uma lista dinâmica para exibir as tarefas cadastradas.
- Botões para editar e excluir cada tarefa.

2. Criar um script JavaScript para:

- Fazer requisições `GET`, `POST`, `PUT` e `DELETE` para o backend usando `fetch()`.
- Atualizar a lista de tarefas automaticamente após cada ação.
- Permitir edição inline das tarefas.

**Dica:** Utilize `DOMContentLoaded` para carregar as tarefas ao abrir a página.

---

## Passo 4: Teste e Ajustes

1. Iniciar o servidor backend e garantir que ele está acessível na porta correta.
  2. Abrir `index.html` no navegador e verificar se as ações CRUD funcionam corretamente.
  3. Corrigir possíveis erros e melhorar a interface conforme necessário.
- 

## Entrega:

- Código fonte do backend e frontend organizados.
- Banco de dados funcional com tarefas armazenadas.
- Testes realizados para garantir que todas as funcionalidades estão operacionais.

**Desafio Extra:**

- Adicionar um campo "status" para marcar tarefas como concluídas.
- Melhorar o design utilizando CSS.
- Implementar um sistema de login para múltiplos usuários.

Boa sorte e bom desenvolvimento!