

# Relatório Técnico: SIG-ProFree

Sistema de Gerenciamento de Projetos para Freelancers

Aluno: Roan Osorio Nunes de Souza  
Matrícula: 202110048811

Dezembro de 2025

## Conteúdo

<b>1</b>	<b>Introdução</b>	<b>3</b>
<b>2</b>	<b>Projeto do Banco de Dados</b>	<b>3</b>
2.1	Modelo Conceitual (Diagrama Entidade-Relacionamento - E-R) . . . . .	3
2.2	Modelo Lógico/Relacional (Esquema das Tabelas) . . . . .	4
2.3	Justificativa das Decisões de Projeto . . . . .	5
<b>3</b>	<b>Arquitetura e Implementação</b>	<b>5</b>
3.1	Tecnologias Utilizadas . . . . .	5
3.2	Comunicação Front-end e Banco de Dados (Integração) . . . . .	5
<b>4</b>	<b>Manual de Uso</b>	<b>5</b>
4.1	Cadastro (CREATE): Formulário de Freelancer . . . . .	6
4.2	Consulta e Exclusão (READ / DELETE): Lista de Clientes . . . . .	6
4.3	Atualização (UPDATE): Formulário de Edição . . . . .	7
<b>5</b>	<b>Conclusão</b>	<b>8</b>
5.1	Principais Dificuldades e Aprendizados . . . . .	8

# 1 Introdução

O presente trabalho tem como objetivo aplicar de forma prática os conceitos de modelagem, projeto e implementação de Sistemas de Banco de Dados (SBD). Para tal, foi desenvolvido o **SIG-ProFree** (Sistema de Gerenciamento de Projetos para Freelancers), uma aplicação completa capaz de gerenciar dados de clientes, projetos, tarefas, freelancers e pagamentos.

A aplicação integra uma interface gráfica de front-end com um banco de dados no back-end, resolvendo o problema da organização de múltiplos projetos e prestadores de serviço, um desafio comum na área de serviços digitais. O projeto atende a todos os requisitos técnicos, incluindo o uso obrigatório de uma interface gráfica (GUI) e a implementação de todas as operações CRUD (Create, Read, Update, Delete).

## 2 Projeto do Banco de Dados

O projeto do banco de dados relacional é o elemento central do sistema, garantindo a persistência e a integridade dos dados.

### 2.1 Modelo Conceitual (Diagrama Entidade-Relacionamento - E-R)

O modelo conceitual foi desenhado utilizando a **Notação de Peter Chen**, conforme o padrão acadêmico. As entidades e os relacionamentos do sistema são apresentados na Figura 1.

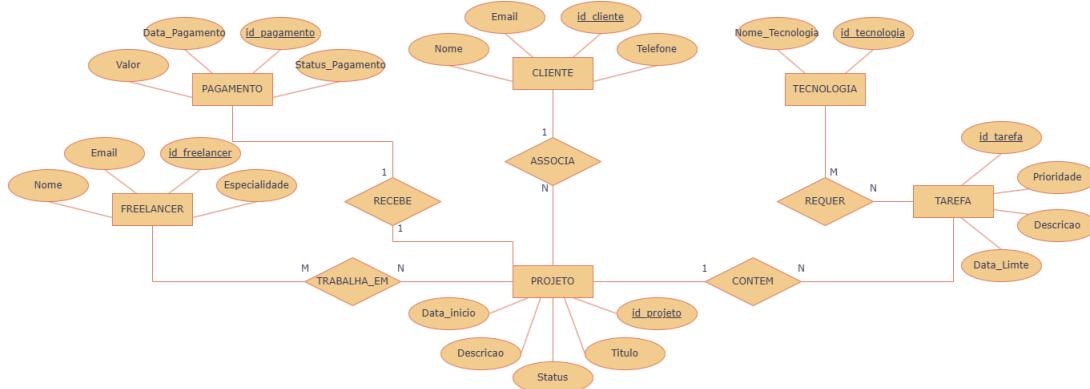


Figura 1: Diagrama Entidade-Relacionamento (E-R) do SIG-ProFree (Notação Chen).

As decisões de modelagem incluem:

- **Relacionamentos 1:N:** CLIENTE → PROJETO e PROJETO → TAREFA.
- **Relacionamentos M:N:** FREELANCER ↔ PROJETO e TAREFA ↔ TECNOLOGIA. Estes exigem tabelas associativas no modelo lógico.
- **Relacionamento 1:1:** PROJETO ↔ PAGAMENTO, garantido pela restrição UNIQUE na Chave Estrangeira.

## 2.2 Modelo Lógico/Relacional (Esquema das Tabelas)

O modelo lógico foi implementado no SGBD MySQL. A seguir, apresentamos o esquema das 8 tabelas, listando suas colunas, tipos de dados e restrições de chave.

### 1. Tabela: cliente

- **id\_cliente:** INT (**PK**)
- **nome:** VARCHAR(100)
- **email:** VARCHAR(100) (**UNIQUE**)

### 2. Tabela: freelancer

- **id\_freelancer:** INT (**PK**)
- **nome:** VARCHAR(100)
- **email:** VARCHAR(100) (**UNIQUE**)
- **Tabela: projeto**
  - **id\_projeto:** INT (**PK**)
  - **titulo:** VARCHAR(100)
  - **id\_cliente:** INT (**FK** → cliente)
- **Tabela: pagamento**
  - **id\_pagamento:** INT (**PK**)
  - **valor:** DECIMAL(10, 2)
  - **id\_projeto:** INT (**FK, UNIQUE** → projeto)
- **Tabela: tarefa**
  - **id\_tarefa:** INT (**PK**)
  - **descricao:** VARCHAR(255)
  - **id\_projeto:** INT (**FK** → projeto)
- **Tabela: tecnologia**
  - **id\_tecnologia:** INT (**PK**)
  - **nome\_tecnologia:** VARCHAR(50) (**UNIQUE**)
- **Tabela Associativa: freelancer\_projeto**
  - **id\_freelancer:** INT (**PK/FK** → freelancer)
  - **id\_projeto:** INT (**PK/FK** → projeto)
- **Tabela Associativa: tarefa\_tecnologia**
  - **id\_tarefa:** INT (**PK/FK** → tarefa)
  - **id\_tecnologia:** INT (**PK/FK** → tecnologia)

## 2.3 Justificativa das Decisões de Projeto

O projeto buscou a conformidade com a **Terceira Forma Normal (3FN)**. O uso de **chaves substitutas (surrogate keys)**, como os IDs inteiros auto-incrementáveis, garante que a chave primária permaneça imutável, simplificando a integridade referencial. Os relacionamentos de Muitos para Muitos (M:N) foram corretamente resolvidos através de tabelas associativas (`freelancer_projeto`, `tarefa_tecnologia`), o que é fundamental para a correção do modelo relacional.

# 3 Arquitetura e Implementação

## 3.1 Tecnologias Utilizadas

A arquitetura do SIG-ProFree é Full-Stack e adota as seguintes tecnologias:

- **Front-end (GUI):** `React.js` (JavaScript), responsável pela interface gráfica e usabilidade.
- **Back-end (API):** `Node.js` e o framework `Express.js` (JavaScript), que expõe os endpoints RESTful.
- **Banco de Dados (SGBD):** `MySQL Server`, utilizado para persistência de dados relacional.

## 3.2 Comunicação Front-end e Banco de Dados (Integração)

A integração é feita por meio de uma arquitetura de três camadas:

- (a) **Front-end (React):** Utiliza a biblioteca `Axios` para enviar requisições HTTP (GET, POST, PUT, DELETE) para a API Node.js, na porta 3001.
- (b) **Back-end (Node.js/Express):** Recebe a requisição e usa o driver `mysql2` para estabelecer o pool de conexões com o MySQL.
- (c) **SGBD (MySQL):** Executa o comando SQL correspondente (e.g., `SELECT * FROM cliente` para a operação de leitura) e devolve o resultado para a API, que o formata em JSON para o React.

# 4 Manual de Uso

A interface gráfica foi projetada para ser intuitiva, utilizando a paleta azul marinho e branco, com cores lógicas para os botões de ação.

#### 4.1 Cadastro (CREATE): Formulário de Freelancer

O formulário de cadastro demonstra a operação **CREATE** na entidade `freelancer`.

The screenshot shows a dark-themed web application interface. At the top center, it says "SIG-ProFree". Below that, in smaller text, is "Sistema de Gerenciamento de Projetos para Freelancers". The main content area has a light gray background and a white border. It contains a form titled "Cadastrar Novo Freelancer (CREATE)". The form fields are: "Nome:" with input "Maria Santos", "Email:" with input "maria@nova.com", and "Especialidade:" with input "Designer UX". At the bottom right of the form is a green button labeled "Cadastrar". Below the form, a green message box displays the success message "Freelancer cadastrado com sucesso."

Figura 2: Interface de Cadastro de Freelancer (CREATE).

#### 4.2 Consulta e Exclusão (READ / DELETE): Lista de Clientes

A seção de Consulta de Clientes demonstra as operações **READ** (carregamento dinâmico dos dados) e **DELETE**.

- O carregamento dinâmico dos dados do banco é feito via requisição GET.
- O botão **Excluir** (vermelho) aciona uma requisição DELETE, demonstrando a Integridade Referencial ao impedir a exclusão de clientes com projetos associados.

**Lista de Clientes (Teste de Conexão READ/DELETE)**

Integração React <--> Node.js <--> MySQL está OK!

ID	Nome	Email	Telefone	Ações
1	TechCorp Inovação	contato@techcorp.com	11987654321	<button style="background-color: red; color: white; border: none; padding: 2px 10px;">Excluir</button>
2	Agência Marketing Digital	contato@agenciamd.com	21912345678	<button style="background-color: red; color: white; border: none; padding: 2px 10px;">Excluir</button>
5	Cliente Teste	teste@excluir.com	9999999999	<button style="background-color: red; color: white; border: none; padding: 2px 10px;">Excluir</button>

Trabalho SBD - UERJ

Figura 3: Interface de Consulta e Exclusão de Clientes (READ/DELETE).

### 4.3 Atualização (UPDATE): Formulário de Edição

O componente de edição demonstra a operação **UPDATE**. O formulário primeiro faz um GET por ID para carregar os dados atuais do freelancer e, em seguida, envia um PUT para a API.

**Editar Freelancer (UPDATE) - ID: 1**

Nome:

E-mail:

Especialidade:

Salvar Alterações

Freelancer atualizado com sucesso!

Figura 4: Interface de Edição de Freelancer (UPDATE).

## 5 Conclusão

O projeto SIG-ProFree atingiu plenamente o objetivo de aplicar os conceitos de modelagem, projeto e implementação de SBD. A aplicação demonstrou a **funcionalidade completa do CRUD** (Create, Read, Update, Delete) em todas as camadas.

### 5.1 Principais Dificuldades e Aprendizados

- **Dificuldade:** A maior dificuldade técnica foi na configuração do ambiente Full-Stack, especialmente na npm e na inicialização do serviço MySQL Server, que gerou erros de ECONNREFUSED.
- **Aprendizado:** Houve um reforço significativo no domínio da **modelagem relacional Peter Chen** e sua conversão correta para o modelo lógico.
- **Robustez:** O sistema provou sua robustez ao validar a **Integridade Referencial** (Chaves Estrangeiras), impedindo operações DELETE que violariam a coerência dos dados.

O trabalho cumpre todos os requisitos do PDF, fornecendo uma aplicação funcional e um relatório detalhado com a justificativa de todas as decisões de projeto e arquitetura.